

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ  
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ  
ГОРОДСКОГО ХОЗЯЙСТВА имени А. Н. БЕКЕТОВА**

**Б. И. Погребняк, Г. В. Высоцкая**

**КОНСПЕКТ ЛЕКЦИЙ  
ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ**

*(для студентов 5-го курса дневной  
и 6-го курса заочной форм обучения магистров  
специальности 8.18010013 «Управление проектами»)*

**ХАРЬКОВ – ХНУГХ – 2013**

**Погребняк Б. И.** Техническое и программное обеспечение: Конспект лекций (для студентов 5-го курса дневной и 6-го курса заочной форм обучения магистров специальности 8.18010013 «Управление проектами») / Б. И. Погребняк, Г. В. Высоцкая; Харьк. нац. ун-т гор. хоз-ва им. А. Н. Бекетова. – Х.: ХНУГХ, 2013. – 90 с.

Авторы: Б. И. Погребняк,  
Г. В. Высоцкая

Конспект лекций построен в соответствии с требованиями кредитно-модульной системы организации учебного процесса и согласован с ориентировочной структурой содержания учебной дисциплины, рекомендованной Европейской Кредитно-Трансферной Системой (ECTS).

Рекомендовано для студентов специальности «Управление проектами».

Рецензент: доцент кафедры Прикладной математики и информационных технологий Харьковской национальной академии городского хозяйства А. Б. Костенко.

Утверждено на заседании кафедры Прикладной математики и информационных технологий.

Протокол № 1 от 30 августа 2013 г.

## ОГЛАВЛЕНИЕ

ЛЕКЦИЯ № 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ.....	4
1. Общая схема обработки информации на компьютере .....	4
2. Понятие информации.....	5
3. Методы получения информации.....	5
4. Свойства информации .....	6
5. Единицы измерения информации .....	7
6. Сигналы, сообщения, данные.....	7
7. Передача информации .....	9
8. Формы представления информации .....	11
ЛЕКЦИЯ № 2 КОМПЬЮТЕР И ЕГО ПАРАМЕТРЫ.....	13
1. Классификация компьютеров.....	13
2. Принципы Джона фон Неймана .....	14
3. Магистрально-модульный принцип построения ЭВМ .....	18
4. Иерархия памяти компьютера .....	35
ЛЕКЦИЯ №3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ.....	36
1. Состав и назначение программного обеспечения .....	36
2. Структура компьютерной программы .....	38
3. Структура операционной системы.....	39
4. Полиморфизм .....	39
ЛЕКЦИЯ №4 СЕТЕВЫЕ ТЕХНОЛОГИИ.....	41
1. Компьютерные сети .....	41
2. Простейшая компьютерная сеть .....	41
3. Классификация компьютерных сетей .....	42
4. Топологии компьютерных сетей.....	44
5. Модель взаимодействия открытых систем ISO/OSI .....	46
6. Понятие протокола и интерфейса .....	51
ЛЕКЦИЯ №5 ВСЕМИРНАЯ КОМПЬЮТЕРНАЯ СЕТЬ ИНТЕРНЕТ.....	52
1. Краткая история создания Интернета.....	52
2. Структура и принципы работы Интернета .....	61
3. Адресация компьютеров в сети.....	64
4. Доменная система имен.....	65
5. Единообразный локатор ресурса.....	74
ЛЕКЦИЯ №6 ОСНОВЫ ВЕБ-ДИЗАЙНА.....	78
1. Назначение языка HTML.....	78
2. Понятие Web-страницы и Web-сайта .....	79
3. История создания языка HTML .....	79
4. Структура HTML-документа.....	80
5. Инструментарий для создания HTML-документов .....	83
СПИСОК ИСТОЧНИКОВ.....	88

# ЛЕКЦИЯ № 1

## ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

### План

1. Общая схема обработки информации на компьютере
2. Понятие информации
3. Методы получения информации
4. Свойства информации
5. Единицы измерения информации
6. Сигналы, сообщения, данные
7. Передача информации
8. Формы представления информации

### 1. ОБЩАЯ СХЕМА ОБРАБОТКИ ИНФОРМАЦИИ НА КОМПЬЮТЕРЕ

Обработка информации с помощью компьютера выполняется точно так же, как фарш в мясорубке. То есть, в мясорубку закладывают ломтики мяса, а после обработки получают то же мясо, только перекрученное – фарш. Аналогично, на вход компьютера подают исходную информацию, а на выходе получают уже обработанную – выходную. Тогда процедура обработки информации на компьютере, в самом общем виде, может быть представлена следующим образом (Рис. 1.1).



Рис. 1.1 – Общая схема обработки информации на компьютере

Из Рис. 1.1 видно, что в этом процессе участвуют три составляющие:

- 1) информация,
- 2) компьютер,
- 3) программы.

Каждая из них далее будет рассмотрена более детально и более глубоко.

## 2. ПОНЯТИЕ ИНФОРМАЦИИ

Термин «информация» происходит от латинского слова «informatio», что означает сведения, разъяснения, изложение, осведомленность, знания и т. д. Само понятие информация является одним из фундаментальных в современной науке вообще, и базовым для информатики в частности. Наряду с такими понятиями, как вещество, энергия, пространство и время, оно рассматривается в качестве важнейшей сущности мира, в котором мы живем. Его нельзя определить через более простые понятия. В математике, например, аналогичными «неопределяемыми» понятиями являются «точка» или «прямая», относительно которых можно сделать некоторые утверждения, но сами они не могут быть определены с помощью более элементарных понятий. Понятие «информация» имеет различный смысл в экономике, науке, технике, различных житейских ситуациях.

В простейшем бытовом значении под информацией понимают сведения об окружающем нас мире и протекающих в нем процессах, воспринимаемые человеком или специальными устройствами. Под информацией в технике понимают сообщения, передаваемые в форме знаков или сигналов. Под информацией в семантическом (смысловом) аспекте понимают сведения, обладающие некоторой новизной. Под информацией в кибернетике, по определению Норберта Винера понимают ту часть знаний, которая используется для ориентирования, активного действия, управления, т. е. в целях сохранения, совершенствования, развития системы. В середине XX века слово «информация» в узком техническом смысле ввел Клод Шеннон применительно к теории связи и передачи сигналов, которая получила название «Теория информации». Под информацией в ней понимают не любые сведения, а лишь те которые, снимают полностью или уменьшают существующую неопределенность относительно некоторого события. То есть, *информация – это мера снятия неопределенности в отношении исхода интересующего нас события.*

*Применительно к компьютерной обработке под информацией понимают:*

- а) некоторую последовательность знаков (кодов, сигналов),*
- б) которая несет некоторую смысловую нагрузку*
- в) и представлена в понятном для компьютера виде.*

## 3. МЕТОДЫ ПОЛУЧЕНИЯ ИНФОРМАЦИИ

Информация о любом событии, объекте, его параметрах, и т. д. может быть получена одним из следующих методов:

**1. Опыт.** Является важнейшим методом получения информации. Он состоит в том, что ежедневно каждый человек, вольно, или невольно, накапливает определенное количество информации – некоторый опыт. В прошлом этот метод был основным и единственным для получения информации в жизни и развитии человека. Множество замечательных

достижений было получено опытным путем, в процессе накопления опыта и выводом определенных умозаключений. Например, какого совершенства достигли древние мастера Китая в изготовлении фарфора, тульские оружейники – в изготовлении оружия.

**2. Эвристический подход.** Его название происходит от слова «эврика», что в переводе с древнегреческого означает «Я нашел!». При эвристическом подходе проводят многократные эксперименты, после которых отбирают наиболее удачные варианты. Поэтому он еще называется «Методом проб и ошибок». Однако этот метод довольно длительный и трудоемкий, а потому недостаточно эффективный.

**3. Целенаправленный поиск.** Он характеризуется тем, что производится не беспорядочный перебор всех возможных вариантов, а анализируются известные достижения в конкретной области, планируются и проводятся опыты. В результате применения целенаправленного поиска человечество создало новые материалы и процессы, ранее не известные в природе. Этому способу получения информации способствует развитие современной техники, которая позволяет обрабатывать огромные объемы информации и получать при этом все новые и новые результаты.

Информация может быть получена так же:

1. путем наблюдения за объектом,
2. эксперимента над ним или
3. путем логического вывода.

Поэтому информация бывает:

1. доопытная (или априорная) и
2. послеопытная, (или апостериорная), полученная в результате проведенного эксперимента.

#### **4. СВОЙСТВА ИНФОРМАЦИИ**

Информация обладает множеством различных свойств. Наиболее существенными из них являются следующие:

**1. Объективность.** Информация объективна, если она не зависит от чьего-либо мнения.

**2. Достоверность.** Информация достоверна, если она отражает истинное положение дел.

**3. Полнота.** Информацию можно считать полной, если ее достаточно для понимания и принятия решения.

**4. Актуальность** – важность, существенность для настоящего времени.

**5. Адекватность** – определенный уровень соответствия создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению.

## 5. ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ

С точки зрения компьютерной обработки наиболее важным свойством информации является ее количество. Определение минимальной единицы измерения количества информации связано с ее вероятностным понятием. Так, если вероятность свершения, или несвершения, какого-либо события одинакова, т. е. равна 0,5 (например, при подбрасывании монеты), то для передачи информации о нем достаточно одного двоичного разряда. Эта единица измерения была названа *бит* (от английских слов binary digital – двоичный разряд). *Один бит – это количество информации необходимое для различения двух равновероятных событий (1 бит = {0, 1})*. Бит также хорошо сочетается с особенностями построения современных цифровых компьютеров, поскольку они также работают по принципу «включено – выключено», «есть сигнал – нет сигнала», «высокий уровень – низкий уровень», «заряжен – разряжен» и т. д.

Количество информации равное 8-ми битам называется *байт* (**1 Байт = 8 бит**). При помощи 8-ми двоичных разрядов можно записать 256 ( $2^8$ ) различных значений от 00000000 до 11111111.

На практике для облегчения работы с большими объемами информации применяют более крупные единицы измерения, такие, как:

- 1 Килобайт** (Кбайт) = 1024 Байт =  $2^{10}$ ;
- 1 Мегабайт** (Мбайт) = 1024 КБайт =  $2^{20}$ ;
- 1 Гигабайт** (Гбайт) = 1024 МБайт =  $2^{30}$ ;
- 1 Терабайт** (Тбайт) = 1024 ГБайт =  $2^{40}$ ;
- 1 Петабайт** (Пбайт) = 1024 ТБайт =  $2^{50}$ .

## 6. СИГНАЛЫ, СООБЩЕНИЯ, ДАННЫЕ

Основной особенностью информации является то, что она является категорией не материальной. Следовательно, для существования и распространения в материальном мире информация должна быть обязательно связана с какой-либо материальной основой – без нее информация не может проявиться, передаваться и сохраняться. Поэтому *материальный объект или среду, которые служат для представления и передачи информации, называют материальным носителем*. Материальным носителем информации может быть бумага, воздух, лазерный диск, электромагнитное поле и пр. *Изменение некоторой характеристики материального носителя, которое используется для представления информации, называют сигналом*. Значение этой характеристики, отнесенное к некоторой шкале измерений, называется *параметром сигнала*. Например, при разговоре материальным носителем является воздух, сигналом – звуковая волна, которая в нем распространяется, а параметрами сигнала – высота и громкость звука.

Однако одиночный сигнал не может содержать много информации. Поэтому для передачи информации используется ряд следующих друг за

другом сигналов. Такая *последовательность сигналов называется сообщением*. Примерами сообщений являются музыкальное произведение, телепередача, команды регулировщика на перекрестке, текст, распечатанный на принтере, данные, полученные в результате работы программы и т. д. *Зарегистрированные каким-либо образом на конкретном материальном носителе сигналы называются данными*. Сам термин «данные» происходит от латинского слова «data» – факт, в то время как «информация» – от латинского «informatio» – сведения, разъяснения, изложение и т. д. Можно сказать, что данные выступают в качестве материальной основы для представления информации. Другими словами, данные служат «исходным сырьем» для получения информации.

Отсюда следует важный вывод о том, что одни и те же данные могут нести различную информацию для различных потребителей. Так, например, данные об антропологических параметрах строения человека для портного, врача и спортивного тренера несут различную информацию. Для портного – это количество необходимого материала для пошива одежды и особенности его раскроя, для врача – отклонения от нормы в пропорциях фигуры и возможные патологии, а для спортивного тренера – пригодность для занятий тем или иным видом спорта и ожидаемые при этом результаты. В то же время сообщения, несущие одну и ту же информацию, образуют целый класс эквивалентных сообщений. Например, прогноз погоды может быть получен по радио, из газеты, по телевидению, из Интернета и т. д.

Исходя из того, что информация фиксируется на материальных носителях в виде данных, и одни и те же данные могут нести различную информацию, следует важное положение о том, что данные могут обрабатываться при помощи различных технических средств. Причем такая обработка может осуществляться формальными методами, поскольку данные не зависят от содержащейся в них информации. Из всех технических средств обработки данных решающая роль, конечно же, принадлежит компьютерам. Данные в компьютере обрабатываются формально, без учета их смыслового содержания, а лишь с использованием математических и логических операций. Оценить смысловое содержание данных в настоящее время может только человек, который находится за пределами системы обработки данных. То есть, обработка данных не всегда является обработкой содержания, а преобразование данных в информацию предполагает наличие соответствующего механизма интерпретации.

*Соответствие между данными и содержащейся в них информацией называется **правилом (методом) интерпретации***. Таким образом, одни и те же данные, по-разному интерпретированные, могут передавать разную информацию. Правило интерпретации может быть известно лишь ограниченному кругу лиц. Связь между данными и информацией особенно отчетлива в криптографии: никто посторонний не должен суметь извлечь из



них информацию. Исходя из факта взаимодействия данных и методов интерпретации, ей можно дать еще одно определение: **информация** – это результат взаимодействия данных и адекватных им методов интерпретации.

## 7. ПЕРЕДАЧА ИНФОРМАЦИИ

Информация не может существовать без наличия источника и приемника (получателя, потребителя) информации. *Источник информации* – это субъект или объект, порождающий информацию и представляющий ее в виде сообщения. *Приемник информации* – это субъект или объект, принимающий сообщение и способный правильно его интерпретировать.

В этих определениях сочетание «субъект» или «объект» означает, что источники и приемники информации могут быть одушевленными (человек, животные) или неодушевленными (технические устройства, природные явления). Для того чтобы объект (или субъект) считался источником информации, он должен не только ее породить, но и иметь возможность создать сообщение. Например, если человек что-то придумал, но держит это в своей голове, он не является источником информации. Однако он им становится, как только изложит свою идею на бумаге (в виде текста, рисунка, схемы и пр.) или выскажет вслух.

В определении приемника информации важным представляется то, что факт приема сообщения еще не означает получение информации. Информация может считаться полученной только в том случае, если приемнику известно правило интерпретации сообщения. Другими словами, понятия «приемник сообщения» и «приемник информации» не тождественны. Например, слыша речь на незнакомом языке, человек оказывается приемником сообщения, но не приемником информации.

Конечным источником и потребителем информации при ее обмене всегда является человек. Следовательно, воспринимать сообщения мы можем только посредством одного из пяти органов чувств, или некоторой их группой. Это не означает, однако, что человек не может использовать для приема и передачи информации какие-то иные сигналы, непосредственно им не воспринимаемые, например радиоволны. В этом случае человек-источник использует промежуточное устройство, преобразующее его сообщение в радиоволны – радиопередатчик, а человек-приемник – другое промежуточное устройство – радиоприемник, преобразующий радиоволны в звук. Такой подход заметным образом расширяет возможности человека в осуществлении передачи и приема информации. Промежуточные устройства-преобразователи получили название *технические средства связи*, а в совокупности с соединяющей их средой они называются *каналом связи* (*каналом передачи информации, информационным каналом*). К ним относятся телеграф, телефон, радио, телевидение, компьютерные телекоммуникации и пр. Передача информации по каналам

связи часто сопровождается воздействием помех, вызывающих искажение и потерю информации (Рис. 1.2).

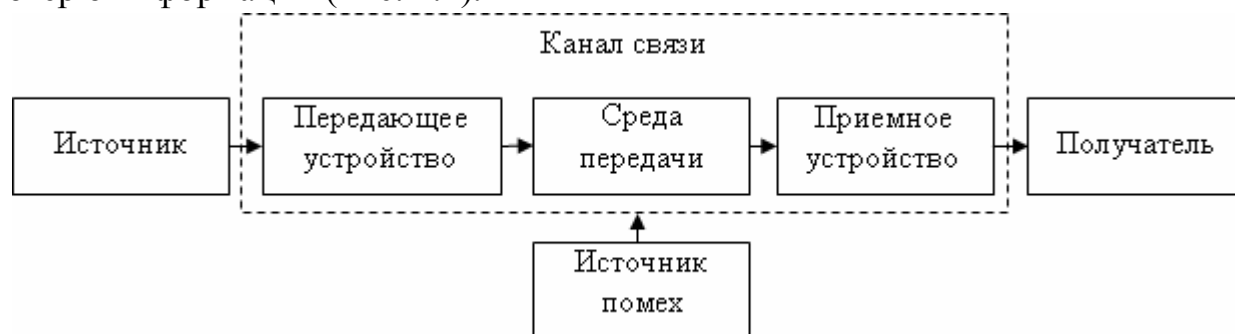


Рис. 1.2. Общая схема передачи информации

Примером канала связи может служить почта. Информация, записанная в виде текста на листе бумаги, помещается в конверт, опускается в почтовый ящик, затем извлекается из него и перевозится в почтовое отделение, где аналогичные отправления сортируются по направлениям назначения. Далее это отправление при помощи различных видов транспорта (автомобилей, поездов, самолетов и т. д.) доставляется в пункт назначения, где аналогичная корреспонденция распределяется по почтовым отделениям, откуда непосредственно доставляются адресату. Таким образом, почтовый канал связи включает в себя: конверт, транспорт, сортировочные машины, и почтовых работников. Информация, переданная при помощи этого канала связи, не искажается.

Другим примером может служить телефон. При этом источником сигнала является говорящий. Кодирующим устройством, преобразующим слова говорящего в электрические импульсы, является микрофон. Канал, по которому передается информация – телефонная сеть. Та часть телефона, которую мы подносим к уху, является декодирующим устройством, в котором электрические сигналы снова преобразуются в звук. Таким образом, информация поступает в «принимающее устройство» – ухо человека на другом конце канала. Канал связи включает в себя телефонные аппараты, соединительные провода, коммутирующие устройства АТС, приемо-передающие устройства и т. д. Особенностью этого информационного канала является то обстоятельство, что при поступлении в него информации в виде звуковых волн (речи), она преобразуется в электрические колебания, и лишь затем – передается. Такой канал называется *каналом с преобразованием информации*.

Еще один пример канала связи – компьютер. Его так же можно рассматривать как информационный канал с преобразованием информации, поскольку, информация, поступающая с внешних устройств (клавиатура, микрофон и т. д.) преобразуется во внутренний код, обрабатывается, а затем снова преобразуется к виду, который может воспринимать устройство вывода (монитор, принтер и т. д.) и передается на них.

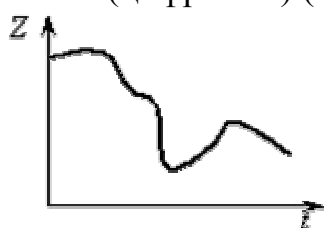
## 8. ФОРМЫ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ

Формы представления информации в современном мире весьма многообразны. Информация может существовать в виде:

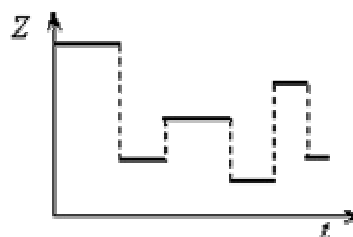
- текстов, рисунков и чертежей;
- световых и звуковых сигналов;
- электрических и нервных импульсов;
- жестов и мимики;
- запахов и вкусов;
- хромосом, передаваемых по наследству свойства организма и т. д.

При этом форма представления информации зависит как от источника, так и от приемника информации, а также канала связи между ними. Так, при общении между людьми обмен информацией происходит в форме, которую могут воспринимать наши органы чувств. Это могут быть речь, мимика, жесты, прикосновения и т. д. При обмене информацией между человеком и техническим устройством могут использоваться звук, различные зрительные образы и т. д. Обмен информацией между техническими устройствами осуществляется при помощи сигналов. При этом по информационным каналам связи могут передаваться два типа сигналов:

1. непрерывные (аналоговые) (Рис. 1.3, а) и
2. дискретные (цифровые) (Рис. 1.3, б).



(а) непрерывные сигналы



(б) дискретные сигналы

Рис. 1.3. Непрерывные и дискретные сигналы

Сигнал называется *непрерывным* (или *аналоговым*), если его параметр может принимать любое значение в пределах некоторого интервала. Примерами непрерывных сигналов являются речь, музыка, изображение, показание термометра (параметр сигнала – высота столба спирта или ртути – имеет непрерывный ряд значений) и пр. Примерно до 70-х годов XX века технические устройства преимущественно работали с аналоговыми сигналами. Это, например, запись и воспроизведение звука, его прием и передача на большие расстояния и т. д.

*Дискретность* сигнала означает, что он представляет собой не непрерывную функцию, а последовательность определенных значений. Дискретный сигнал называется *цифровым*, если его параметр может принимать конечное число значений в пределах некоторого пронумерованного диапазона. Сообщение, передаваемое с помощью таких сигналов, то же называется дискретным, и информация, передаваемая источником в этом случае, также является дискретной. Пример дискретного сообщения – процесс чтения книги,

информация в которой представлена текстом, т. е. дискретной последовательностью отдельных знаков (букв).

Принципиальным и важнейшим различием непрерывных и дискретных сигналов является то, что дискретные сигналы можно обозначить, т. е. приписать каждому из них некоторый *знак*, который будет отличать данный сигнал от других сигналов. *Знаком называется элемент некоторого конечного множества отличных друг от друга сущностей*. Природа знака может любой – жест, рисунок, буква, сигнал светофора, определенный звук и т. д. Его природа определяется носителем данных и формой представления информации в них.

Вся совокупность знаков, используемых для представления дискретной информации, называется *набором знаков* – т. е., набор есть дискретное множество знаков. Набор знаков, в котором установлен порядок их следования, называется *алфавитом*. То есть, *алфавит – это упорядоченная совокупность знаков*. Порядок следования знаков в алфавите называется *лексикографическим*. Благодаря этому порядку между знаками алфавита можно установить отношения «больше–меньше». Для двух знаков «а» и «я» принимается, что  $a < я$ , если порядковый номер у *а* в алфавите меньше, чем у *я*. Примером алфавита так же может служить совокупность арабских цифр 0, 1...9. С помощью такого алфавита можно записать любое целое число в системах счисления от двоичной до десятичной. Поскольку при передаче сообщения параметр сигнала должен меняться, очевидно, что минимальное количество различных его значений равно двум и, следовательно, *алфавит может содержать минимум два знака*. Такой алфавит называется *двоичным*.



Представляется важным еще раз подчеркнуть, что *понятия знака и алфавита можно отнести только к дискретным сообщениям*.

Любое непрерывное сообщение может быть представлено непрерывной функцией, заданной на некотором интервале. Непрерывное сообщение можно преобразовать в дискретное. Эта процедура называется *дискретизацией* (или *оцифровкой*). Из бесконечного множества значений параметра непрерывного сигнала выбирается их определенное число, которое приближенно может характеризовать остальные значения. Для этого диапазон изменения функции разбивается на отрезки равной длины и на каждом из этих отрезков значение функции принимается постоянным и равным (например, среднему значению на каждом отрезке). В итоге получается конечное множество чисел. Таким образом, любое непрерывное сообщение может быть представлено как дискретное, иначе говоря, последовательностью знаков некоторого алфавита.

Возможность дискретизации непрерывного сигнала с любой желаемой точностью (для возрастания точности достаточно уменьшить шаг) принципиально важна с точки зрения информатики. Потому, что подавляющее большинство современных компьютеров – это цифровые устройства, т. е. внутреннее представление информации в них дискретно. А дискретизация входной непрерывной (аналоговой) информации позволяет сделать ее пригодной для компьютерной обработки.

## ЛЕКЦИЯ № 2

### КОМПЬЮТЕР И ЕГО ПАРАМЕТРЫ

#### План

1. Классификация компьютеров
2. Принципы Джона фон Неймана
3. Магистрально-модульный принцип построения ЭВМ
4. Иерархия памяти компьютера

#### 1. КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ

Компьютеры могут быть классифицированы по самым разным признакам. Например, по назначению и области применения, по мощности и габаритам, по внутренней архитектуре и т. д. Однако, наиболее существенной из них является классификация по *принципу действия*. По этому признаку все компьютеры могут быть разделены на следующие три категории:

- *аналоговые* – вычислительные машины непрерывного действия, работающие с информацией, представленной в аналоговой форме (в виде непрерывного ряда значений какой-либо физической величины);
- *цифровые* – вычислительные машины дискретного действия, работающие с информацией, представленной в цифровой (дискретной) форме;
- *гибридные* – вычислительные машины смешанного действия, позволяющие обрабатывать информацию, представленную как в цифровой, так и в аналоговой форме.

Для изучения различных процессов и явлений в науке и технике широко используют *метод аналогий*, когда вместо оригинала объекта А изучают его модель Б. Такое замещение возможно при наличии *соответствий (аналогий)* между объектом-оригиналом А и объектом-моделью Б. При этом различают аналогии двух видов: *физические* и *математические*.

В случае *физического моделирования* объект Б является уменьшенной (реже – увеличенной) копией оригинала, и при изменении оригинала требуется менять и модель, что является недостатком этого метода. В случае *математического моделирования* оригинал и модель имеют сходство математических описаний, хотя их физическая природа может быть совершенно разной.

Для исследования математических моделей используются аналоговые вычислительные машины – АВМ. Такие компьютеры (электронные вычислительные машины – ЭВМ) представляют собой совокупности соединенных определенным образом так называемых *решающих блоков*, которые могут складывать сигналы, вычитать, интегрировать, выполнять любые функциональные преобразования. Подготовка задачи для решения на АВМ, т. е. *программирование*, включает два этапа:

1. соединение (коммутация) решающих блоков в соответствии с задачей, и
2. установка коэффициентов масштабирования физических величин.

*Достоинствами АВМ* являются:

- предельно существующее в природе *быстродействие* – практически равное скорости распространения света (электромагнитной волны), и
- *простота программирования*, которая сводится к соединению необходимым образом решающих блоков.

К недостаткам АВМ относятся:

- *ручная коммутация* решающих блоков (программирование), связанная с реализацией алгоритма, и
- *низкая точность* решения задач, которая зависит от точности электронных элементов (резисторов, конденсаторов и т. д.), которые составляют решающие блоки.

АВМ предназначены, в основном, для решения задач, описываемых системами дифференциальных уравнений: исследования поведения подвижных объектов, моделирования поведения процессов и систем, оптимального управления. Однако они не могут эффективно решать задачи, связанные с длительным хранением и обработкой больших объемов информации, которые легко решают цифровые вычислительные машины – ЦВМ. Поэтому со временем они были практически полностью вытеснены цифровыми ЭВМ.

Наиболее распространенными ЭВМ в мире, на сегодняшний день, конечно же, являются цифровые – ЦВМ (ЭЦВМ). Он обрабатывает информацию, представленную в цифровом виде – в виде набора нулей и единиц. Основными достоинствами ЦВМ являются:

- *гарантированная точность* результата, зависящая только от диапазона представления данных, который может быть установлен как аппаратным, так и программным способом (8, 16, 32, 64, 128 и т. д. двоичных разряда);
- *универсальность (гибкость)* – способность решать практически любую задачу – для этого необходимо лишь заменить программу, которая будет управлять работой такой ЦВМ.

Гибридные вычислительные машины (ГВМ, аналого-цифровые – АЦВМ) сочетают достоинства аналоговых и цифровых вычислительных машин. Они чаще всего применяются в узкоспециализированных областях, таких как медицина и навигация, а также выступают в роли аналого-цифровых и цифро-аналоговых преобразователей.

## 2. ПРИНЦИПЫ ДЖОНА фон НЕЙМАНА

В 1946 году трое ученых – Артур Беркс (англ. Arthur Burks), Герман Голдстейн (англ. Herman Goldstine) и Джон фон Нейман (англ. John von Neumann) – опубликовали статью «Предварительное рассмотрение логического конструирования электронного вычислительного устройства» [1]. Она была написана на основании критического анализа конструкции ЭВМ ENIAC (Electronic Numeral Integrator And Computer),



Джон фон Нейманн (1903-1957)

созданной под руководством Джона Мокли (англ. John William Mauchly) и Дж. Преспера Экерта (англ. John Presper Eckert). Джон фон Нейман подключился к проекту **ENIAC** позднее в качестве консультанта-математика. По сути, фон Нейман обобщил научные разработки и открытия многих других ученых и придал инженерным решениям Мокли и Экерта академический, научный стиль. То есть, авторство концепции в действительности принадлежит коллективу авторов, и ее правильнее было бы называть *принципы Мокли-Экерта-фон Неймана*. Но, поскольку, имя Джона фон Неймана было достаточно широко известно в науке того времени, то это отодвинуло на второй план его соавторов, и данные идеи получили название *принципы фон Неймана*.

В целом же они сводятся к следующему:

- **Принцип структуры (устройства) ЭВМ.** Компьютер фон-неймановского типа состоит из следующих отдельных блоков (Рис. 2.1) и которые выполняют такие функции:

- 1) **Арифметико-Логическое Устройство (АЛУ, англ. ALU – Arithmetic and Logic Unit)** – выполняет все арифметические и логические операции, необходимые для обработки информации, хранящейся в **ЗУ**.

- 2) **Устройство Управления (УУ, англ. CU – Control Unit)** – организует процесс выполнения программ, управляет и контролирует работу всех остальных узлов ЭВМ.

- 3) **Запоминающее Устройство (ЗУ, Оперативное Запоминающее Устройство – ОЗУ, запоминающее устройство произвольного доступа, оперативная память, англ. RAM – Random Access Memory)** – хранит программы и данные.

- 4) **Устройства Ввода-Вывода (УВВ, англ. IOU – Input-Output Units)** – служат для ввода программ и данных в компьютер, и вывода результатов.

Устройство управления (**УУ**) и арифметико-логическое устройство (**АЛУ**) со временем были объединены в единый блок – *Центральный Процессор (ЦП, англ. CPU – Central Processing Unit)*, являющийся основным, и единственным, преобразователем информации в компьютере.

- **Принцип двоичного кодирования.** Для представления *данных и команд* используется двоичная система счисления. Преимущество перед десятичной системой счисления заключается в том, что устройства ЭВМ можно делать достаточно простыми, арифметические и логические операции в двоичной системе счисления также выполняются достаточно просто. В дальнейшем ЭВМ стали обрабатывать и нечисловые виды информации – текстовую, графическую, звуковую и другие, но двоичное кодирование данных по-прежнему составляет основу любого компьютера. Команды и данные в двоичной системе счисления образуют единицы информации, которые называются *машинными словами* (4, 8, 16, 32, 64, 128 и т. д. бит). Именно такими фрагментами происходит обмен информацией между **ЗУ** и **ЦП** за одно обращение.

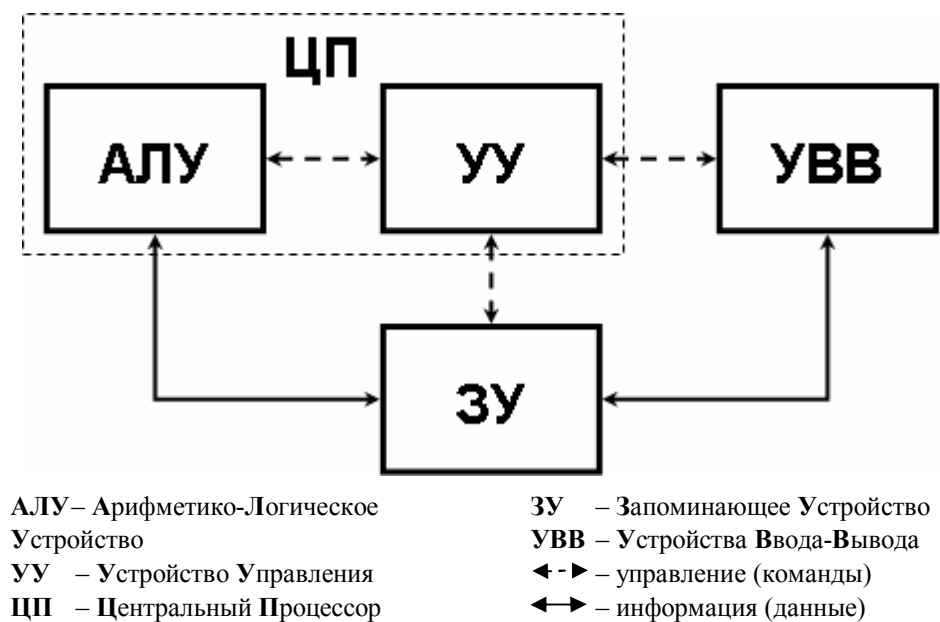


Рис. 2.1. Структура ЭВМ фон Неймана

• **Принцип однородности памяти (хранимой программы).**

Как программы (команды), так и данные хранятся в одной и той же памяти. Поэтому, ЭВМ не различает, что хранится в конкретный момент времени в данной ячейке памяти. Разнотипные слова различаются лишь по способу использования, но не по способу кодирования – т. е., над командами можно выполнять такие же действия, как и над данными. Отсутствие принципиальной разницы между программой и данными открывает целый ряд возможностей. Например, команды одной программы могут быть получены как результат исполнения другой программы. На этом принципе основаны методы трансляции – перевода текста программы с языка программирования высокого уровня на язык конкретной ЭВМ. Созданием компьютера с хранимой в памяти программой было положено начало тому, что мы сегодня называем *программированием*.

• **Принцип адресуемости памяти.** Структурно **ЗУ** состоит из пронумерованных ячеек памяти. Номер ячейки памяти называется ее *адресом*. **ЦП** в произвольный момент времени имеет возможность доступа к любой из них.

• **Принцип автоматического (программного) управления.**

1) При помощи устройства ввода программа заносится (вводится, загружается и т. д.) в **ЗУ** компьютера.

2) Все команды, составляющие программу, выполняются **ЦП** автоматически (без вмешательства человека), начиная с первой, строго последовательно, друг за другом. Адрес очередной команды вычисляется как адрес текущей команды, увеличенный на ее длину.

3) Если же после выполнения текущей команды нужно перейти не к следующей, а к какой-то другой команде, используются команды передачи управления – условного и безусловного переходов. В таких командах в



качестве операнда указывается адрес ячейки памяти, содержащей необходимую команду (куда необходимо передать управление). Команды передачи управления позволяют реализовать переход к любому участку кода, обеспечивая тем самым возможность организации ветвления и циклов, что позволяет использовать одни и те же последовательности команд в программе много раз (для циклов) или выполнять различные участки кода в зависимости от выполнения определенных условий (ветвление) и т. д. (Этот принцип был сформулирован задолго до фон Неймана Адой Лавлейс и Чарльзом Бэббиджем, однако, он логически включен в фон-неймановский набор как дополняющий предыдущий принцип.)

4) Результаты выполнения программы выводятся на устройство вывода.

5) Выборка команд из памяти прекращается после достижения и выполнения команды «Стоп».

6) После этого компьютер переходит в режим ожидания сигнала от устройства ввода.

Самым же главным следствием этих принципов можно назвать то, что теперь программа уже не была постоянной частью компьютера (как например, у калькулятора). Программу стало возможно легко изменить. А вот аппаратура, конечно же, остается неизменной. То есть, главная задача, которая стояла перед «пионерами» вычислительной техники была успешно решена. Для сравнения, управление (программирование) компьютера **ENIAC** осуществлялось с помощью специальной коммутационной панели. Для того чтобы задать такую программу приходилось в течение нескольких часов, или даже дней, устанавливать нужным образом многочисленные переключатели, тумблеры, разъемы и другие коммутирующие элементы. Худшим же из всех недостатков была ужасающая ненадежность компьютера – за день работы успевало выйти из строя около десятка электронных ламп – т. е. собственно расчет не мог продолжаться более нескольких минут.

Со временем в английском языке для программного обеспечения было выбрано (а точнее, создано) довольно удачное слово *SoftWare* (буквально – «мягкое изделие»), которое подчеркивает равнозначность программного и аппаратного обеспечения («железа» – *HardWare* – «жесткое изделие»), и вместе с тем говорит о его гибкости, способности модифицироваться, приспосабливаться, развиваться. Введение этих терминов было связано с необходимостью провести четкую грань между командами-инструкциями, управляющими компьютером, и его физическими компонентами или аппаратным обеспечением, которое, собственно, и составляет компьютер. Поэтому переключение компьютера с построения рисунка на разработку стандартного контракта или с разработки архитектурного проекта на создание карты погоды земного шара стало осуществляться простым изменением последовательности команд, управляющих его работой, т. е. программ. Техническое (аппаратное) обеспечение (*hardware*) вместе с программным обеспечением (*software*) составляют полную вычислительную систему (ВС). На практике, и в обиходе, довольно часто всю ВС

называют компьютером. Путаницы с использованием данного термина, как правило, не происходит – его значение всегда можно определить из контекста, в котором он используется.

Как только на практике были воплощены принципы фон Неймана – родился современный компьютер общего назначения, который «стоит на следующих трех китах»:

1. *центральный процессор,*
2. *оперативная память и*
3. *внешние устройства.*

С тех пор прошло более полувека, но выдвинутые положения сохраняют свою актуальность и сегодня – была создана архитектура ЭВМ, во многих чертах сохранившаяся и до настоящего времени. Большинство современных компьютеров в основных чертах соответствуют принципам, изложенным фон Нейманом, хотя со временем, естественно, они были адаптированы к текущим реалиям. Например, процесс выполнения программы может быть прерван для выполнения неотложных действий, связанных с поступившим сигналом от внешнего устройства компьютера – прерывания.

Компьютер фон-неймановского типа не является идеальным (хотя практически получился довольно «живучим») и не лишен определенных недостатков. Основной из его недостатков является продолжением его достоинств. Поскольку архитектура ЭВМ является не монолитной, а состоит из отдельных блоков (**ЦП**, **ЗУ** и **УВВ**), то всегда существовала (и существует поныне) проблема согласования (синхронизации) быстродействия работы его узлов. Такое различие в быстродействии может достигать до нескольких порядков. На практике эта проблема в различных ЭВМ решается по-разному. Например, память компьютера делается иерархической («многоярусной»), или для хранения программ и данных отводятся отдельные запоминающие устройства (Гарвардская архитектура). Наиболее же распространенной архитектурой ЭВМ на сегодняшний день (ЭВМ 3-го и 4-го поколений: на интегральных схемах – ИС и на больших интегральных схемах – БИС, соответственно) является шинная (или магистрально-модульная), в которой (в том числе и) для обеспечения необходимого быстродействия используются специализированные процессоры – контролеры внешних устройств (устройства управления).

### **3. МАГИСТРАЛЬНО-МОДУЛЬНЫЙ ПРИНЦИП ПОСТРОЕНИЯ ЭВМ**

Благодаря усилиям многих фирм-разработчиков ЭВМ, и IBM в первую очередь, идеи фон Неймана со временем были реализованы в виде широко распространенного в наше время принципа *открытой архитектуры*. В соответствии с этим принципом, компьютер не является единым и неделимым устройством, а состоит из независимо изготовленных модулей (частей). Причем, методы сопряжения этих модулей с компьютером не являются секретом фирмы-производителя и доступны всем желающим. Модульный принцип построения ЭВМ позволяет потребителю самому

подобрать нужную ему конфигурацию (по принципу детского конструктора) и производить, при необходимости, его модернизацию (апгрейд, англ. upgrade – «повышать уровень»). Основой открытой архитектуры является *системная магистраль (шина)* – некая виртуальная общая дорога (или жила, или канал), в которую выходят все выводы ото всех узлов ЭВМ.

Структурная (логическая) схема современного компьютера общего назначения магистрально-модульной организации представлена на Рис. 2.2.

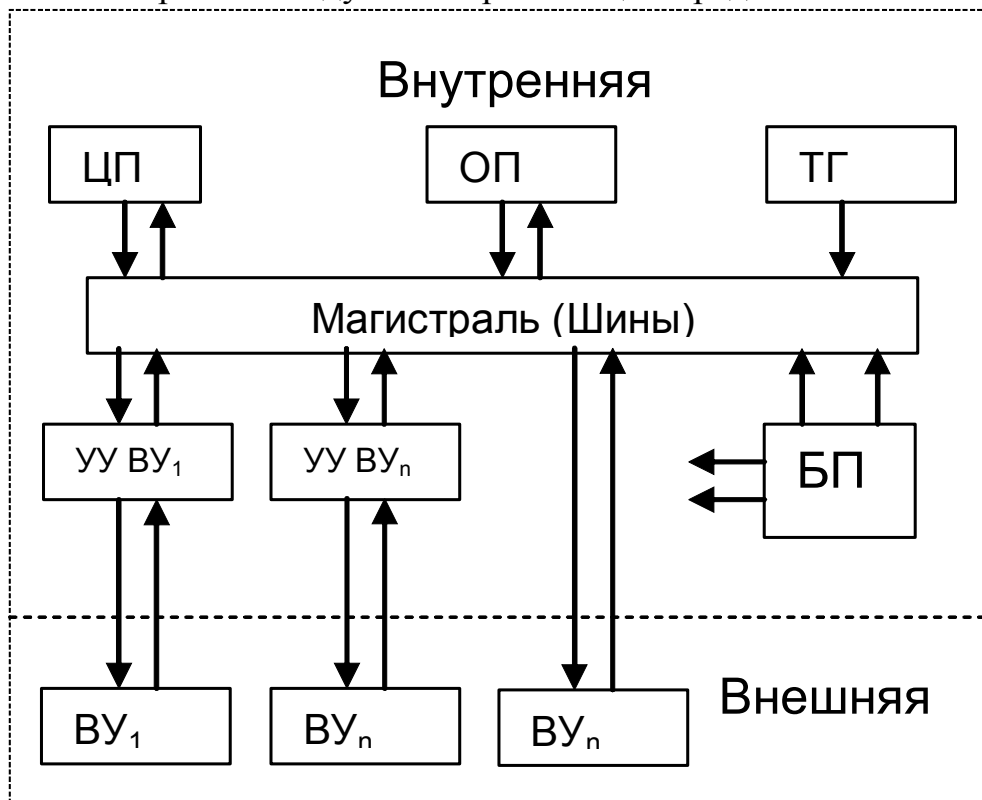


Рис. 2.2 – Магистрально-модульная архитектура современного компьютера общего назначения

Он состоит из двух классов устройств:

1. *внутренних (центральных)*, которые обеспечивают весь процесс обработки информации, и

2. *внешних (периферийных)*, предназначенных для передачи информации из внешней среды в компьютер и обратно, а также для длительного ее хранения.

*Центральный процессор (ЦП)* является основным устройством любой ЭВМ. Он определяет все остальные характеристики компьютера. Именно в нем происходит обработка информации. Совокупность всех операций, которые может выполнять процессор, образует так называемую *систему (набор, архитектуру) команд*.

В современных персональных компьютерах разными фирмами-производителями устанавливаются процессоры двух основных наборов команд:

- **CISC** (сокр. от англ. *Complete Instruction Set Computer*) – процессоры с полным набором инструкции и

- **RISC** (сокр. от англ. *Reduced Instruction Set Computer*) – процессоры с сокращенной системой команд.

Весь модельный ряд процессоров фирмы Intel, устанавливаемых в персональные компьютеры типа IBM PC, имеют систему команд CISC, а процессоры Motorola, используемые фирмой Apple для своих персональных компьютеров Macintosh, имеют набор команд RISC. Обе архитектуры имеют свои преимущества и недостатки. Так CISC-процессоры имеют обширный набор команд (около 400), из которых программист может выбрать команду, наиболее подходящую ему в данном случае. Недостатком этой архитектуры является то, что большой набор команд усложняет внутреннее устройство процессора, тем самым, увеличивая его стоимость, увеличивается время исполнения команд, команды имеют различную длину и время исполнения.

Система команд RISC-процессоров отличается относительной простотой: коды инструкций имеют четкую структуру, как правило, фиксированной длины. В результате аппаратная реализация такой архитектуры позволяет с небольшими затратами выполнять эти инструкции за минимальное время. Нередко слова «сокращенный набор команд» понимаются как минимизация количества инструкций в системе команд. В действительности, инструкций у многих RISC-процессоров больше, чем у CISC-процессоров. На самом деле, термин «сокращенный» в названии описывает тот факт, что сокращен объем (и, соответственно, время) работы, выполняемый каждой отдельной инструкцией – как максимум один цикл доступа к памяти, – тогда как сложные инструкции CISC-процессоров могут требовать сотен циклов доступа к памяти для своего выполнения. К недостаткам RISC-архитектуры можно отнести то, что если требуемой команды в наборе нет, программист вынужден реализовать ее с помощью нескольких команд из имеющегося набора, увеличивая тем самым размер программного кода.

Формально, все процессоры Intel являлись CISC-процессорами. Однако, начиная с Intel 486, они являются CISC-процессорами с RISC-ядром. Такая архитектура получила название **MISC** (англ. *Multipurpose Instruction Set Computer*) – процессоры с многоцелевыми командами. Они еще называются процессорами с микропрограммным управлением. Эти процессоры непосредственно перед исполнением преобразуют CISC-инструкции в более простой набор внутренних инструкций RISC. При этом одна команда CISC может порождать несколько RISC-команд. Это потребовалось для увеличения скорости обработки CISC-команд, так как известно, что любой CISC-процессор уступает RISC-процессору по количеству выполняемых операций в единицу времени. В итоге, такой подход позволил поднять общую производительность таких CISC-процессор, а также использовать старое программное обеспечение без изменений. После того, как процессоры CISC-архитектуры были переведены на систему команд RISC, можно смело сказать, что большинство существующих ныне процессоров основаны на архитектуре RISC.

Однако помимо перечисленных выше систем команд процессоров существуют и другие. Наиболее известные из них, например, такие:

- **MISC** (сокр. от англ. *Minimal Instruction Set Computer*) – *процессоры с минимальным набором команд* (не путать с MISC – Multipurpose Instruction Set Computer), являются вырожденным (предельным случаем) системы команд RISC.

- **VLIW** (сокр. от англ. *Very Long Instruction Word*) – *очень длинное командное слово*. Характеризуются тем, что одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно. То есть, обеспечивается выполнение группы непротиворечивых команд за один цикл работы процессора.

- **EPIC** (сокр. от англ. *Explicitly Parallel Instruction Cod*) – *процессоры с явным параллелизмом команд*. EPIC позволяет микропроцессору выполнять инструкции параллельно, опираясь на работу компилятора, а не выявляя возможность параллельной работы инструкций при помощи специальных схем. Она является дальнейшим развитием VLIW-архитектуры. Архитектура EPIC реализована фирмой Intel в 64-разрядном процессоре *Intel Itanium* и получила название **IA-64** (сокр. от англ. Intel Architecture – 64).

Процессоры, имеющие одинаковые системы команд, так же *совместимы и на программном уровне*. Это значит, что программа, написанная для одного процессора, будет выполняться и на всех остальных процессорах с такой же системой команд. Процессоры, имеющие различные системы команд, как правило, полностью, или частично, несовместимы на программном уровне.

Некоторые процессоры, имеющие ограниченную совместимость, образуют так называемые *семейства (клоны) процессоров*. Так, например, все процессоры корпорации Intel, которые устанавливаются в компьютеры типа IBM PC (и имеют систему команд CISC) относятся к так называемому семейству **x86**. Родоначальником этого семейства был 16-разрядный Intel 8086, на базе которого и были собраны первые персональные компьютеры – знаменитые IBM PC. Название семейства образовано от двух цифр, которыми заканчивались названия процессоров Intel ранних моделей – 8086, 80186, 80286, 80386, 80486. Впоследствии выпускались так же процессоры Intel Pentium и Intel Core. Самое последнее семейство процессоров Intel Core имеет несколько модификаций: Intel Core i3, Intel Core i5 и Intel Core i7. Помимо Intel, архитектура x86 также была реализована в процессорах других производителей: AMD, VIA, Transmeta, IDT и др. В настоящее время для этой архитектуры существует еще одно название – **IA-32** (сокр. от англ. Intel Architecture – 32). Все процессоры этой архитектуры обладают программной совместимостью по принципу «снизу вверх».

Принцип совместимости «снизу вверх» – это пример довольно удачного технического решения, когда каждый последующий процессор «понимает» все команды своих предшественников (но не наоборот). Благодаря такой совместимости на современном компьютере можно выполнять программы,

созданные ранее для любого из предшествующих компьютеров, принадлежащего к той же аппаратной платформе. Примером полной несовместимости по системе команд с процессорами семейства x86 являются процессоры корпорации Motorola (поскольку они являются полностью RISC-процессорами). На их основе компанией Apple создано целое семейство компьютеров Macintosh. Хотя по назначению и классу решаемых задач они точно такие же, как и IBM PC, но программы, созданные для IBM PC на компьютерах Macintosh работать не будут, как, впрочем, и наоборот. Программная совместимость процессоров является одной из важнейших их характеристик. На Рис. 2.3 представлен график изменения стоимости производства программ и компьютеров во всем мире с течением времени.

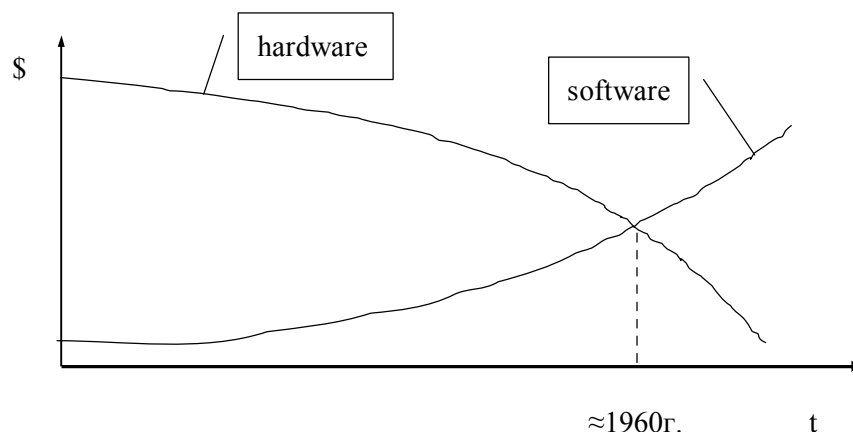


Рис. 2.3. Тенденция изменения стоимости программных и аппаратных средств компьютера

Он демонстрирует то, что с течением времени затраты на создание программных средств увеличиваются, а компьютеров — уменьшаются. Из этого следует, что в подавляющем большинстве случаев, если некоторая программа перестала удовлетворять по быстродействию, гораздо дешевле заменить компьютер на более совершенный, нежели переписывать существующую программу. Соответственно, при замене одного компьютера на другой, с отличной от первого системой команд, придется нести и дополнительные (многократно превышающие стоимость самого компьютера) затраты по созданию для него необходимых программ.

С остальными устройствами компьютера, и в первую очередь с оперативной памятью, центральный процессор связан с помощью так называемых *шин* — систем параллельных сигнальных проводников. К основным типам шин центрального процессора относятся:

1 *Адресная шина* — служит для указания процессором адреса ячейки памяти, с которой будет начинаться запись или считывание данных.

2 *Шина данных* — через которую осуществляется обмена информацией между оперативной памятью компьютера и центральным процессором.

3 *Шина команд* — используется для пересылки команд из оперативной памяти в центральный процессор.

Следующей по важности характеристикой центрального процессора после системы команд является его *разрядность*. Она указывает количество бит данных, которое центральный процессор способен обработать, хранить и передавать на другие устройства за один такт (одновременно). Разрядность центрального процессора определяется разрядностью регистров, в которые помещаются обрабатываемые данные. Например, если регистры имеют разрядность 2 Байта, то разрядность процессора равна 16 бит ( $2 \times 8$ ); если 4 Байта, то 32 бита; а если 8 Байт – то 64 бита.

Все процессоры семейства x86 (IA-32) 32-разрядные. Но уже к началу 2000-х годов стало очевидно, что 32-битное адресное пространство архитектуры x86 ограничивает производительность приложений, работающих с большими объемами данных. Оно позволяет процессору осуществлять непосредственную адресацию лишь 4 ГБайт данных. И поэтому при некоторых обстоятельствах эта характеристика может иметь решающее влияние на производительность всего компьютера и стать его «узким местом».

29 мая 2001 года компаниями Intel и Hewlett-Packard впервые совместно была представлена основанная на наборе команд EPIC новая 64-разрядная архитектура под названием IA-64. Она была воплощена в процессорах Itanium и Itanium 2. Для обеспечения обратной совместимости со старыми приложениями, использующими 32-разрядный код, в IA-64 был предусмотрен режим эмуляции. Однако на практике данный режим работы оказался чрезвычайно медленным, и поэтому архитектура IA-64 не получила широкого распространения.

Компания AMD предложила альтернативное решение данной проблемы – вместо того, чтобы изобретать совершенно новую систему команд, было предложено ввести 64-разрядное расширение к уже существующей 32-разрядной архитектуре x86. Первоначально новая архитектура называлась *x86-64*, позже она была переименована в *AMD64*. В различных источниках она имеет и другие названия. Наиболее распространенные из них это: *x64*, *IA-32e*, *Intel 64* и *EM64T*.

Начиная с 2003 года все процессоры, выпускаемые обеими мировыми лидерами индустрии – Intel и AMD, являются исключительно этой архитектуры. То есть, 64-разрядными, с обратной совместимостью работы в 32-битном режиме. Поэтому актуальность использования несовместимой с ними платформы IA-64 где-либо, кроме серверов, пропала вовсе.

Пользователи иногда путают архитектуру Intel 64 с IA-64, ошибочно пытаясь установить на свои компьютеры программы для этой архитектуры, и затем обнаруживают, что они не запускаются. Во избежание подобных ошибок, следует помнить, что Intel 64 и IA-64 – это совершенно разные, несовместимые между собой, микропроцессорные архитектуры.

На скорость обработки информации внутри центрального процессора влияет его *внутренняя тактовая частота*. В современных процессорах семейства Intel Core i3/i5/i7 она достигает порядка нескольких ГГц. Для сравнения, процессор Intel 8086, на основе которого был построен первый компьютер IBM PC, работал на частоте всего лишь 4.77 МГц!

*Оперативная память* (**ОП**, или *оперативное запоминающее устройство* – **ОЗУ**, **RAM** – Random Access Memory – память произвольного доступа) – это устройство, хранящее ту информацию, с которой компьютер работает непосредственно в данное время (исполняемая программа, часть необходимых для нее данных, некоторые управляющие программы и т. д.).

Организация оперативной памяти компьютера чем-то напоминает систему абонентских почтовых ящиков, с которыми мы имеем дело в повседневной жизни, в том смысле, что они содержат и хранят объекты. То есть, объекты содержатся во множестве отдельно адресуемых *ячеек памяти*, каждая из которых имеет свой персональный *адрес*. При обращении к памяти необходимо указать адрес соответствующей ячейки. Но на этом сходство и заканчивается.

Способность же оперативной памяти компьютера хранить данные основывается совсем не на том, что объекты физически присутствуют в ней, а на том, что состояние самой ячейки соответствует некоторому закодированному представлению объекта. Следовательно, ячейка должна принимать некоторое количество *дискретных состояний*. Трудно изготовить компоненты, способные принимать любое из множества отличных друг от друга состояний и оставаться в нем сколь угодно долго. Однако можно без особых затруднений создавать запоминающие элементы, которые имеют только два различных состояния. Такие элементы называются *двоичными запоминающими элементами*. Если сгруппировать *n* двоичных запоминающих элементов, то эта группа может принимать  $2^n$  различных состояний. Если эту группу рассматривать как некоторую неделимую компоненту, то она представляет собой ячейку памяти с  $2^n$  возможными состояниями. Один двоичный разряд в такой группе, который может принимать одно из двух значений – 0 или 1, называется *бит*. Группа, состоящая из восьми ( $n = 8$ ) двоичных разрядов (бит) называется *байт*. Таким образом, байт может принимать одно из  $2^8 = 256$  различных значений – от 0 до 255.

Каждая ячейка оперативной памяти компьютера адресуется с помощью целого положительного числа. Максимальный объем оперативной памяти компьютера в первую очередь зависит от разрядности процессора. Так, в процессорах семейства x86 (IA-32) непосредственно можно обратиться к ячейке с максимальным адресом  $2^{32} = 4\,294\,967\,296$ , т. е. максимальный объем оперативной памяти такого компьютера не может превышать 4.3 ГБайт. Однако это отнюдь не означает, что именно столько памяти непременно должно быть в компьютере. Предельный размер оперативной памяти, которую можно установить в компьютере, иногда зависит от так называемого *чипсета* (англ. *chip set* – *микروпроцессорного комплекта*) – набора вспомогательных микросхем, управляющих обменом данных внутри компьютера. Помимо ограничений, накладываемых на размер оперативной памяти компьютера центральным процессором и другими вспомогательными микросхемами, он зависит также от характера задач, решаемых на компьютере, а также от ее удельной стоимости. Тогда, с учетом всех этих ограничений, можно



сформулировать следующее правило – для *компьютера общего назначения лишней памяти не бывает никогда*.

Объем информации, пересылаемый за одно обращение, между оперативной памятью и центральным процессором зависит от разрядности центрального процессора. Обычно это одно из значений следующего ряда: **8, 16, 32, 64, 128** и т. д. бит. Из него видно, что *минимально адресуемой величиной оперативной памяти компьютера является байт*. При этом за одно обращение современные процессоры пересылают не 8, а значительно больше бит информации. Даже если необходимо переслать один байт информации, все равно пересылается больше. Такая процедура пересылки информации принята потому, что между центральным процессором и оперативной памятью находится несколько уровней буферной, так называемой, *кэш-памяти* (англ. *cache* – прятать), которая работает гораздо быстрее ОЗУ, поскольку в подавляющем большинстве программ данные в оперативной памяти располагаются подряд. При этом если данные уже находятся в кэш-памяти их пересылка в регистры центрального процессора (и, соответственно, из регистров в кэш-память) выполняется на несколько порядков быстрее, чем прямо из оперативной памяти. За счет применения кэш-памяти и «широкого» считывания информации из оперативной памяти (и, соответственно, записи) значительно возрастает общее быстродействие компьютера.

При работе с оперативной памятью следует четко отличать адрес ячейки (адрес байта) и содержание ячейки (ее значение). Адрес (номер) ячейки всегда один и тот же, а вот ее содержимое может меняться в диапазоне от 0 до 255. Это напоминает упоминавшуюся выше систему абонентских почтовых ящиков, адреса которых всегда остаются постоянными, а вот содержимое в каждый отдельный промежуток времени может быть различным.

Принципиальной особенностью оперативной памяти является ее способность хранить информацию только во время работы компьютера. В процессе выполнения программы содержимое ячеек памяти постоянно меняется – в одни из них информация заносится из процессора как результаты выполнения некоторых арифметических или логических операций, в другие информация пересылается из иных участков памяти, а в третьи – из внешней среды через *внешние устройства*. По выражению известного американского программиста Питера Нортона, оперативная память – это черновик, где временно записываются программы, данные, и результаты обработки – полуфабрикаты. Для пользователя же эти результаты обретают законченный вид, если они оформлены и выведены в определенном виде, на принтер, на экран монитора на звуковые колонки и т. д. После загрузки в оперативную память новой программы, ее прежнее содержимое замещается новым, а после выключение компьютера – и вовсе пропадает.

Существует много разновидностей оперативной памяти, но с точки зрения физических принципов ее работы различают два основных вида:

1 **DRAM** – Dynamic RAM – динамическая память, и

## 2 **SRAM** – Static RAM – статическая память.

Ячейки динамической памяти (DRAM) представляют собой микроконденсаторы, которые на своих пластинах способны хранить электрический заряд. Это наиболее распространенный, экономичный и доступный тип памяти. Однако этот тип памяти обладает и определенными недостатками, которые заключаются в следующем:

- Как при заряде, так и при разряде конденсаторов неизбежны переходные процессы, в связи с чем, запись и считывание информации осуществляется в несколько раз медленнее, чем в памяти типа SRAM.

- С течением времени заряды ячеек имеют свойство рассеиваться в пространстве, причем весьма быстро, что может привести к потере информации. Для борьбы с этим явлением приходится постоянно *регенерировать (освежать, подзаряжать)* ячейки оперативной памяти, что является дополнительным источником непроизводительных расходов ресурсов компьютера, и как следствие – снижение быстродействия.

Ячейки статической памяти (SRAM) представляют собой электронный *триггер*, имеющий два устойчивых состояния. Поскольку в триггере хранится не заряд, а состояние (включен/выключен), то этот тип памяти обеспечивает более высокое быстродействие, хотя за счет того, что каждая ячейка содержит несколько транзисторов и других электронных компонентов, он сложнее и, соответственно, дороже.

Микросхемы динамической памяти в компьютерах общего применения используются, как правило, в качестве основной оперативной памяти, микросхемы же статической памяти – в качестве вспомогательной (промежуточной, *кэш*-) памяти, предназначенной для увеличения быстродействия компьютера.

Основной движущей силой развития оперативной памяти является все возрастающее увеличение быстродействия компьютеров и их центральных процессоров. Было разработано несколько типов памяти DRAM: PM DRAM – page mode DRAM – страничная память, FPM DRAM – fast page mode DRAM – быстрая страничная память, EDO DRAM – extended data out DRAM – память с усовершенствованным выходом, SDR SDRAM – Single Data Rate Synchronous Dynamic Random Access Memory – синхронная память с одинарной скоростью передачи данных. Наибольшее распространение в качестве оперативной памяти в компьютерах общего применения получила память типа DDR SDRAM – double data rate SDRAM – память с удвоенной скоростью передачи данных. В ней данные передаются по 2 бита на один синхроимпульс – как по фронту, так и по срезу тактового импульса. Память DDR SDRAM также постоянно совершенствуется. Было выпущено несколько поколений этой памяти: DDR, DDR2, DDR3 и DDR4, которые отличаются не только емкостью и скоростью, но и потреблением электроэнергии.

Элементы динамической памяти для компьютеров общего применения конструктивно выполняются в виде так называемых *модулей*, которые

вставляются в соответствующие разъемы шин. При этом наибольшее распространение получили следующие два основных исполнения:

- **DIMM** – Dual In-line Memory Module – модуль памяти с двухрядным расположением выводов. Они применяются в настольных компьютерах в виде 184-контактных (DDR) или 240-контактных модулей (DDR2 и DDR3).

- **SO-DIMM** – Small outline DIMM – аналоги модулей DIMM в компактном исполнении. Используются в портативных компьютерах: ноутбуках и нетбуках.

Основными характеристиками модулей оперативной памяти являются их емкость и время доступа. Современные DIMM- и SO-DIMM-модули имеют емкость в несколько Гбайт. Время доступа показывает, сколько времени необходимо для обращения к ячейке памяти – чем оно меньше, тем быстрее может выполняться обмен с памятью. Время доступа измеряется в *наносекундах* – *нс*. Его типичное значение для DIMM-модуля порядка 4 нс., что обеспечивает пиковую скорость передачи данных около 17000 МБ/с.

**Шины** (англ. *bus*) – это среда обмена информацией между всеми устройствами компьютера. Физически шина представляет собой:

- ряд проводников (линий),
- разъемов (*слотов*) и
- электронных схем (*чипсет*),

а логически – это определенный набор (семейство) правил обмена информацией. Каждое такое семейство правил имеет свое, уникальное, имя – название шины. Синонимом термина «шина» является «магистраль».

Выделенная из общей шины группа линий, выполняющих одинаковые функции, также называется шиной. Это такие шины, как:

- *адресная шина*, которая используется для адресации устройств компьютера,
- *шина данных*, предназначенная для передачи данных и
- *шина команд (управления)*, служащая для передачи команд (управляющих сигналов).

Неоднозначностей между понятием общей шины и ее составляющими, как правило, не возникает, поскольку смысл всегда понятен из контекста.

Основная характеристика шины – ее пропускная способность, которая самым существенным образом влияет на производительность всего компьютера в целом.

В персональных компьютерах общего применения различают два типа шин:

- **локальная**, обеспечивающая обмен между центральными устройствами компьютера, такими как процессор и оперативная память, и
- **расширения**, предназначенная для подсоединения внешних устройств и являющаяся производной от локальной.

Одной из первых, широко используемой, локальной шиной была *Multibus I*, первоначально разработанная фирмой Intel для компьютеров на базе процессора Intel 8086. Ее адресные шины мультиплексируются с шинами данных. Или, проще говоря, одни и те же линии попеременно используются для передачи адресов и данных. Позднее эта шина была доработана и для

использования в компьютерах на базе процессора Intel 80286. Для компьютеров с 32-разрядным процессором, начиная с Intel 80386, использовалась локальная шина *Multibus 2*, также разработанная фирмой Intel. Начиная с процессоров Intel Pentium Pro, используется локальная шина *FSB – Front Side Bus*. Ее частота является одним из основных потребительских параметров компьютера, и может быть одной из ряда: 66, 100, 133, 200, 266 и более МГц.

С самой шиной расширения пользователь компьютера непосредственно дела не имеет. Он «общается» с ней посредством гнезд (разъемов, слотов) расширения, вставляя в них различные платы расширения – графические, звуковые, сетевые и т. д. Хотя за время, истекшее с момента появления первого компьютера, было разработано довольно много типов шин расширения, и связанных с ними слотов, все они разрабатывались в рамках определенных стандартов. Именно благодаря этой стандартизации слоты расширения стали открытой системой, допускающей быструю модернизацию и расширение функций компьютера. Уже при выпуске первых компьютеров фирма IBM сделала доступными для всех технические справочники с описанием назначения контактов и всех необходимых технических деталей слотов расширения. Для фирм-производителей это явилось стимулом к разработке новых и более совершенных плат расширения. Этот процесс усовершенствования не прекращается и в настоящее время – пока не видно ни его конца, ни насыщения рынка.

Разнообразные современные шины расширения, разработанные с использованием последних достижений науки и техники, далеко ушли от своих предшественников. Но гнезда расширения, обеспечивающие работу этих шин, по своему функциональному назначению и внешнему исполнению, мало чем отличаются от стандартных слотов первых компьютеров фирмы IBM. При этом обычно не возникает никаких проблем, связанных с выбором требуемой платы расширения любого функционального назначения – нужно лишь выяснить, какая шина, или шины, используются в данном компьютере.

Исторически самой первой шиной расширения семейства компьютеров IBM PC была **PC-BUS** – Personal Computer Bus – шина персонального компьютера. Она и центральный процессор синхронизировались от одного тактового генератора с частотой 4.77 МГц. При этом максимальная скорость передачи данных не превышала 2 МБайт/с. На сегодняшний день наиболее производительной шиной расширения является **PCI Express**, постепенно вытесняющая шины **PCI** и **AGP**.

**PCI** – Peripheral Component Interconnect – межсоединение периферийных компонентов – первую версию которой в 1992 году представила корпорация Intel. Intel так же объявила, что стандарт шины будет открытым и создала организацию PCI Special Interest Group (PCI SIG), которая стала заниматься развитием стандарта. Благодаря этому любой заинтересованный разработчик получил возможность создавать устройства для шины PCI без необходимости приобретения лицензии.

По своей сути шина PCI – это интерфейс локальной шины, связывающей процессор с оперативной памятью, в который «врезаны» разъемы для подключения внешних устройств. В данном случае локальная шина и шины расширения четко отделены друг от друга. Связь между ними осуществляется через так называемые *мосты (bridges)*, функции которых выполняют вспомогательные микросхемы (*chipset*). При этом имеется два моста: *северный* и *южный*. Северный мост обеспечивает подключение ЦПУ к более «быстрым» узлам, таким как ОЗУ и графический контроллер. А южный – к более медленным – контролерам магнитных дисков, контролерам шин PCI, PCI Express, USB и т. д. Названия мостов «северный» и «южный» объясняется представлением архитектуры чипсета в виде карты, на которой более быстрый мост расположен сверху – на севере – а более медленный – внизу – на юге. В этом случае за пакетную передачу данных (*burst mode*) отвечает не процессор, а мосты. Процессор может продолжать свою работу, когда шина PCI записывает данные в память или считывает их из нее. То же самое происходит и при обмене данными между двумя PCI-платами расширения. Благодаря этой возможности обеспечивается заметное повышение производительности в многозадачных режимах с помощью специально разработанных для этих целей программ-драйверов.

Фактически шина PCI в свое время заменила собой другие 32-разрядные шины, такие как **EISA** и **VLB**. В тоже время она могла одновременно применяться с 16-разрядной шиной **ISA** под широко распространенные 16- и 8-разрядные платы расширения. Шина PCI могла работать на частотах 33 и 66 МГц, обеспечивая при этом пиковую пропускную способность в 133 и 266 МБайт/с, соответственно. Однако, в связи с развитием компьютерной графики и разработкой шины **AGP**, она перестала удовлетворять новым, повышенным требованиям к видеокартам и перестала использоваться для их установки.

**AGP** – Accelerated Graphics Port – ускоренный графический порт – была разработанная компанией Intel в 1996 году как специализированная шина для видеокарт. Ее пропускная способность в режиме 8-и кратного умножения (спецификация AGP 3.0) достигает 2 ГБайт/с. Современные видеокарты требуют большой мощности, более 40 Вт, которую шина AGP дать не может – так появилась спецификация **AGP Pro** с дополнительными разъемами питания.

**PCI Express (PCIe, PCI-E)** – компьютерная шина, использующая протокол, основанный на последовательной передаче данных. Официально первая базовая спецификация PCI Express была представлена компанией Intel в июле 2002 года. Она призвана заменить параллельную шину PCI и ее расширенный и специализированный вариант AGP. Развитием стандарта PCI Express, как и прежде, занимается организация PCI SIG.

Несмотря на похожие наименования, шины PCI и PCI Express имеют мало общего. Протокол передачи данных шины PCI параллельной, а PCI Express – последовательной, который предоставляет возможность масштабирования (x1, x2, x4, x8, x16 и x32). Шина PCI Express x1 работает на частоте 2.5 ГГц и обеспечивает пропускную способность в 250 Мбит/с для каждого устройства в одном

направлении. При этом соединение (англ. *link* – связь, соединение) между двумя устройствами PCI Express состоит из одной (x1) или нескольких (x2, x4, x8, x16 и x32) двунаправленных последовательных линий. Каждое из устройств должно поддерживать, по крайней мере, соединение по одной линией (x1).

Шина PCI Express нацелена на использование только в качестве локальной шины. Так как программная модель PCI Express во многом унаследована от PCI, то существующие системы и контроллеры могут быть доработаны для использования шины PCI Express заменой только физического уровня, без доработки программного обеспечения. Высокая пиковая производительность шины PCI Express позволяет использовать ее вместо шин AGP и тем более PCI. Де-факто PCI Express заменила эти шины в персональных компьютерах.

В ноябре 2010 года организация PCI SIG, которая занимается стандартизацией технологии PCI Express, объявила о принятии спецификации PCIe 3.0. При этом общая пропускная способность шины PCIe 3.0 в конфигурации x16 достигает 32 Гбит/с и она по-прежнему сохраняет обратную совместимость с предыдущими версиями PCI Express. Ожидается, что PCI Express 4.0 может быть стандартизирована до 2015 года, и будет работать в два раза быстрее, чем PCIe 3.0.

**USB** – Universal Serial Bus – универсальная последовательная шина – последовательный интерфейс передачи данных для среднескоростных и низкоскоростных периферийных устройств. Удобство этой шины расширения состоит в том, что она практически исключает конфликты между различным оборудованием, позволяет подключать и отключать устройства в «горячем режиме», т. е. без выключения компьютера. Контроллер шины расширения USB, как правило, интегрирован в микросхему южного моста, хотя может быть исполнен и в отдельном корпусе. Соединение контроллера с внешними устройствами происходит через USB-концентратор (англ. *hub* – хаб, разветвитель). В силу того, что USB-шина имеет древовидную топологию, концентратор самого верхнего уровня называется корневым (*root hub*). Он встроен в USB-контроллер и является его неотъемлемой частью. Разработка спецификаций на шину USB производится в рамках международной некоммерческой организации USB Implementers Forum (USB-IF), объединяющей разработчиков и производителей оборудования с шиной USB. Спецификация шины USB 1.0 была представлена 15 января 1996 года. Окончательный вариант стандарта USB 3.0 появился в 2008 году. Он повышает максимальную скорость передачи информации до 5 Гбит/с – что на порядок больше 480 Мбит/с, которые может обеспечить USB 2.0.

Блокам  $УУ В У_1 \div УУ В У_n$  на Рис. 2.2 соответствуют «устройство управления внешним устройством<sub>1</sub>»  $\div$  «устройство управления внешним устройством<sub>n</sub>». Устройства управления внешними устройствами (*контроллеры, карты, адаптеры, платы расширения* и т. д.) предназначены для управления работой внешних устройств. Наличие интеллектуальных контроллеров стало важной отличительной чертой машин третьего и четвертого поколений. Так,

подключение отдельных модулей компьютера к магистрали на физическом уровне осуществляется с помощью контроллеров, а на программном обеспечивается драйверами. Контроллер можно рассматривать как специализированный процессор, управляющий работой «вверенного ему» внешнего устройства по специальным встроенным программам обмена. Такой процессор имеет собственную систему команд. Например, контроллер накопителя на магнитных дисках (дисковод) умеет позиционировать головку на нужную дорожку диска, читать или записывать сектор, форматировать дорожку и т. п. Результаты выполнения каждой операции заносятся во внутренние регистры памяти контроллера и могут быть в дальнейшем прочитаны центральным процессором. Таким образом, наличие интеллектуальных внешних устройств может существенно изменять идеологию обмена. Центральный процессор при необходимости произвести обмен выдает задание на его осуществление контроллеру. Дальнейший обмен информацией может протекать под руководством контроллера без участия центрального процессора. Последний получает возможность «заниматься своим делом», т. е. выполнять программу дальше (если по данной задаче до завершения обмена ничего сделать нельзя, то можно в это время решать другую). Описанную схему легко пополнять новыми устройствами – это свойство называют *открытостью архитектуры*. Для пользователя открытая архитектура означает возможность свободно выбирать состав внешних устройств для своего компьютера, т. е. конфигурировать его в зависимости от круга решаемых задач.

Как правило, каждому внешнему устройству соответствует свое устройство управления; хотя существуют и многоканальные контролеры, каждый из которых управляет работой несколько внешних устройств. Конструктивно устройство управления может быть выполнено как в виде небольшой интегральной микросхемы, так и в виде довольно сложной платы расширения. Так работой клавиатуры и мыши в персональном компьютере управляют простые контролеры, а монитором – достаточно сложное устройство – *видеокарта (видеоадаптер)*.

Видеокарта совместно с монитором образует так называемую *видеосистему* персонального компьютера. Для увеличения производительности видеокарт используются так называемые *видеоускорители (видеоакселераторы)*. Они берут на себя часть функций по построению изображения, разгружая при этом центральный процессор и, соответственно, уменьшая количество пересылаемой информации. Такими операциями являются: рисование линий и многоугольников, закрашивание определенным цветом указанных многоугольников, перемещение фрагментов изображения и т. д.

Различают два типа акселераторов: 2D (2-dimension – 2-х мерные) и 3D (2-dimension – 2-х мерные). Первые из них оптимизированы для работы с «плоскими» изображениями, например, для офисных приложений. Вторые же ориентированы на работу с компьютерными играми и профессиональными программами обработки трехмерной графики. Обычно в этих случаях

используют разные математические принципы ускорения графических операций, но существуют акселераторы, обладающие функциями и двухмерного и трехмерного ускорения.

Питание всех блоков и узлов центральной части компьютера осуществляется при помощи *блока питания* (блок *БП* на Рис. 2.2). Он осуществляет преобразование энергии внешней сети в необходимые каждому устройству напряжения. Основная характеристика блока питания – мощность. Блоки питания современных компьютеров имеют мощности порядка сотен ватт. Блок питания всегда должен иметь некоторый запас по мощности. Это необходимо по двум причинам:

1. Для обеспечения надежной работы компьютера при не стабильной питающей сети. Так, кратковременные падения напряжения в сети ниже допустимого уровня при не достаточной мощности блока питания могут вызывать плохо диагностируемые сбои в его работе.

2. При модернизации (апгрейде) компьютера, особенно при замене таких энергоемких устройств как центральный процессор или видеокарта, мощности имеющегося блока питания может не хватить. В этом случае придется нести дополнительные затраты, связанные и с заменой блока питания.

Блоки  $ВУ_1 \div ВУ_n$  на Рис. 2.2 – это «внешнее устройство<sub>1</sub>» ÷ «внешнее устройство<sub>n</sub>». Они служат для:

- ввода,
- вывода и
- длительного хранения информации.

Основными устройствами ввода информации в компьютер являются клавиатура и мышь, а вывода – дисплей и принтер.

От параметров дисплея во многом зависит производительность и комфортность работы на компьютере. Основными потребительскими параметрами дисплея являются:

- способ получения изображения,
- размер по диагонали и
- соотношение сторон.

По способу получения изображения наиболее распространенными являются следующие две группы мониторов:

1. на основе электронно-лучевой трубки (*ЭЛТ*, англ. *CRT – Cathode Ray Tube*) и
2. жидкокристаллические (*ЖК*, англ. *LCD – Liquid Crystal Display*)

Размер экрана монитора по диагонали принято измерять в дюймах. По соотношению сторон наиболее распространенными группами мониторов являются следующие:

- стандартные (4:3),
- широкоформатные (16:9 или 16:10) и
- другие (например, 5:4)



Помимо общих для всех мониторов характеристик, дисплеи на жидких кристаллах имеют свои, специфические, характеристики. Так, для них большое значение имеют такие параметры, как:

- Разрешение – горизонтальный и вертикальный размеры, выраженные в пикселях. В отличие от ЭЛТ-мониторов, ЖК имеют одно фиксированное разрешение, остальные достигаются интерполяцией.
- Время отклика – минимальное время, необходимое пикселю для изменения своей яркости. Оно характеризует степень инерционности вывода изображения.
- Угол обзора – это угол, при котором падение контраста достигает заданного уровня.

Принтер (устройство печати) предназначен вывода информации на бумажный носитель, или получения «твердой копии» (англ. *hard copy*). По способу получения изображения наиболее распространенными типами принтеров являются:

1. лазерные и
2. струйные.

Основными потребительскими характеристиками принтеров являются следующие:

1. Формат бумаги, на которую осуществляется вывод, например, А4, А3 и т. д.
2. Качество печати, или разрешающая способность, которая измеряется в количестве точек на дюйм (англ. *dpi – dots per inch*). Модели среднего класса струйных и лазерных принтеров обеспечивают разрешение до 600 dpi, а высшего – до 2400÷2800 dpi.

3. Производительность, или скорость печати. Она измеряется в количестве отпечатанных страниц в минуту – (англ. *ppm – pages per minute*).

4. Удельная стоимость печати одного листа. Она состоит из первоначальной стоимости самого принтера и стоимости расходных материалов. Стоимость же расходных материалов, в свою очередь, состоит из стоимости красящего вещества и стоимости печатающей головки, или светочувствительного барабана, которые постепенно изнашиваются, и подлежат замене. В начальный период эксплуатации наиболее экономичными являются относительно дешевые струйные принтеры. При больших же объемах печати более экономичными становятся лазерные принтеры, поскольку ресурс светочувствительного барабана достаточно высок, и одной заправки картриджа тонером хватает на большое количество отпечатков.

Внешние запоминающие устройства (ВЗУ) предназначены для длительного хранения и переноса информации между компьютерами. По способу регистрации (записи) информации наиболее распространенными типами ВЗУ являются:

- магнитные, и
- оптические.

Всякое внешнее запоминающее устройство состоит из 2-х частей:

- носителя (диск, лента и т. д.) на котором, собственно, и происходит регистрация информации, и

- накопителя (привод, драйвер, дисковод, «карман» и т. д.), в котором осуществляются операции записи-считывания информации.

Конструктивно накопитель и носитель информации в некоторых случаях могут быть выполнены в виде единого устройства, например, жесткие магнитные диски, или в виде двух отдельных компонент, например, привод лазерного диска и сам лазерный диск.

По способу организации доступа к информации ВЗУ делятся на устройства:

1. последовательного и
2. прямого

доступа. В устройствах последовательного доступа для того, чтобы прочитать (или записать)  $n$ -ю порцию данных необходимо предварительно прочитать (или записать)  $n-1$  таких блоков. В устройствах прямого доступа запись (или считывание) требуемого блока данных выполняется непосредственно. Всякое устройство прямого доступа всегда можно использовать и в последовательном режиме, но обратное – никогда. Наиболее распространенными устройствами последовательного доступа являются магнитные ленты, а прямого – диски (магнитные и оптические).

Наиболее важными характеристиками любого внешнего запоминающего устройства, точки зрения пользователя, являются следующие три:

1. емкость,
2. скорость записи-чтения информации, и
3. удельная стоимость хранения единицы данных, которая оценивается как отношение стоимости накопителя и носителя информации к их емкости.

По человеческим меркам устройства ввода-вывода работают очень быстро, однако, по сравнению со скоростью обмена информацией между другими устройствами компьютера, их скорость крайне низкая.

*Магнитные* диски обеспечивают память большой емкости, позволяют подолгу и без значительных затрат хранить большие объемы информации, будь то программы или данные. Если оперативную память компьютера можно сравнить с памятью человека, то магнитные диски (или внешние запоминающие устройства) сопоставимы с библиотекой. Человеку доступна хранящаяся в библиотеке информация, но прежде, чем ею воспользоваться, он должен ее прочесть. Точно так же информация, хранящаяся во внешней памяти, должна быть перенесена в оперативную память, чтобы с ней мог работать центральный процессор. Основными характеристиками магнитных дисков являются емкость и скорость записи-чтения информации.

В современном компьютере в качестве основного хранилища информации наиболее часто применяются жесткие диски («винчестеры», hard disks). Жесткий диск – это герметически запаянный металлический корпус, в который помещены головки записи-чтения и несколько, посаженных на одну ось, дисков. Связь с остальными электронными устройствами компьютера осуществляется через специальный разъем, расположенный на его корпусе.

Можно спорить о том, находится ли информация, хранящаяся на магнитных дисках, «внутри» компьютера или нет. Если, например, информация записана на оптический диск, то его можно вынуть из дисководов и полностью отделить от компьютера. Таким образом, магнитные диски могут рассматриваться не как память внутри компьютера, а как устройства ввода-вывода. По сравнению с процессором и оперативной памятью они работают довольно медленно, но все же намного быстрее других типов устройств ввода-вывода; в этом их преимущество перед последними.

#### **4. ИЕРАРХИЯ ПАМЯТИ КОМПЬЮТЕРА**

Принцип иерархической организации памяти был сформулирован в связи с тем, что с самого первого компьютера с сохраняемой программой существовало несоответствие между быстродействием арифметического устройства и оперативной памяти. Противоречия бы не существовало, если бы выполнить память на тех же элементах, что и арифметическое устройство. Но такая память получалась слишком дорогой, кроме того, непомерно увеличивалось количество электронных элементов, что заметно снижало надежность компьютера. Иерархическое построение оперативного запоминающего устройства позволяет иметь быстродействующую память небольшого объема только для данных и команд, подготовленных к непосредственному выполнению. Все остальное хранится в запоминающем устройстве более низкого уровня. Для этого стали использоваться появившиеся вскоре магнитные носители информации.

Память современного персонального компьютера разделена на три уровня, в зависимости от емкости, скорости доступа и цены. Это:

- внешняя память или ВЗУ; наиболее емкая и наиболее медленная; информация в ней может храниться сколь угодно долго, даже после выключения питания.
- оперативная память или ОЗУ предназначена для хранения программ и данных, необходимых при их выполнении; занимает промежуточное положение по емкости и скорости доступа; информация «стирается» после выключения питания.
- сверхоперативная память – регистры центрального процессора; емкость регистра общего назначения соответствует длине машинного слова, их количество ограничено, а скорость обмена информацией – максимальная; информация в них так же «стирается» после снятия питания.

# ЛЕКЦИЯ №3

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

### План

1. Состав и назначение программного обеспечения
2. Структура компьютерной программы
3. Структура операционной системы
4. Полиморфизм

### 1. СОСТАВ И НАЗНАЧЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Секрет столь широкого использования компьютеров в различных областях человеческой деятельности кроется в программном обеспечении, которое позволяет использовать одну и ту же машину, построенную, из кремния и стали, для решения огромного множества самых разнообразных задач. В отличие от первых программ, вводившихся в машину посредством утомительного переключения множества тумблеров и цифро-наборных устройств, программное обеспечение, вдохнувшее жизнь в современный компьютер, записывается, как правило, на магнитных дисках и вступает в работу сразу после включения машины. Однако по существу команды компьютера, хотя они и задаются теперь гораздо более удобным способом, заметных изменений не претерпели. Любой компьютер должен разложить задание на последовательность машинных команд, а затем выполнять их одну за другой.

В английском языке для программного обеспечения (ПО) выбрано (а точнее, создано) довольно удачное слово *software* (буквально – «мягкое изделие»), которое подчеркивает равнозначность программного и аппаратного обеспечения («железки» – *hardware* – «жесткое изделие»), и вместе с тем говорит о его гибкости, способности модифицироваться, приспосабливаться, развиваться. Введение этих терминов было связано с необходимостью провести четкую грань между командами-инструкциями, управляющими компьютером, и его физическими компонентами или аппаратным обеспечением, которое, собственно, и составляет компьютер. Поэтому переключение компьютера с построения рисунка на разработку стандартного контракта или с разработки архитектурного проекта на создание карты погоды земного шара осуществляется простым изменением последовательности команд, управляющих его работой, т. е. программ.

Прежде, чем обсуждать состава и назначения программного обеспечения, введем еще два термина: *программист* и *пользователь*. Если программа сложная, в ее создание вовлекается большая группа людей, выполняющих различные функции, то мы будем употреблять общий термин «программист»

для обозначения человека, который проектирует, записывает и совершенствует новую программу, или модифицирует старую. Человек, для которого пишется программа и который, вероятно, будет снабжать ее входными данными и использовать полученные результаты, называется пользователем. И хотя один и тот же человек может быть и пользователем, и программистом, важно различать эти две ипостаси.

Общий термин программное обеспечение (ПО, *software*) применяется для обозначения программ, используемых в компьютерных системах. Все ПО можно классифицировать по множеству самых разнообразных признаков. Наиболее существенными из них являются классификации по:

- назначению, или месту использования в технологическом процессе обработки информации, и
- виду (типу) *лицензии*.

В зависимости от назначения программы подразделяются на два больших класса, которые называются, соответственно:

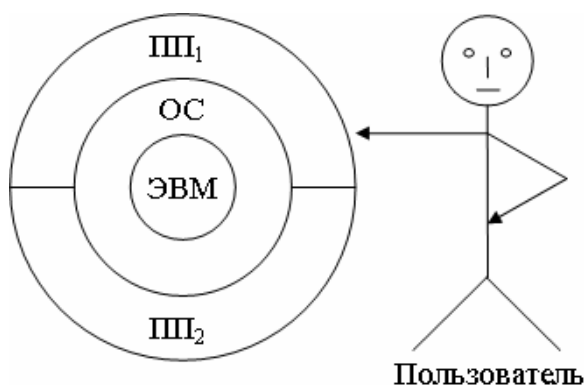
1. прикладное программное обеспечение (ППО) и
2. базовое программное обеспечение (БПО).

Прикладное программное обеспечение образуют программы для пользователей и осуществляющие информационные процессы, которые нужны пользователям.

Центральное место в базовом программном обеспечении занимает специальная совокупность программ, называемая *операционной системой*, которая обеспечивает взаимодействие:

1. человека,
2. компьютера и
3. прикладных программ.

Такое взаимодействие выполняется в соответствии с Рис. 3.1.



ЭВМ – электронная вычислительная машина (компьютер)

ОС – операционная система

ПП<sub>1</sub>, ПП<sub>2</sub> – прикладная программа<sub>1</sub>, прикладная программа<sub>2</sub>

Рис. 3.1. Взаимодействие пользователя, компьютера и программ

Операционная система дает указание компьютеру, как интерпретировать команды и данные, как распределять аппаратные ресурсы для выполнения заданий и как управлять периферийными устройствами (например, принтером или дисплеем). Она также обеспечивает возможность непосредственного взаимодействия человека и компьютера, выполняя такие действия, как хранение программ и данных на внешних запоминающих устройствах.

Если рассматривать операционную систему как «режиссера» компьютерного действия, то прикладные программы играют роль «артистов». Именно благодаря таким программам, как текстовые процессоры, игры и электронные таблицы, компьютер приобретает удивительную разносторонность.

Понятие «базовое программное обеспечение» включает также такие программы, как трансляторы и утилиты. Транслятор – это программа, которая принимает в качестве исходных данных другую программу, написанную на так называемом языке программирования высокого уровня, отдаленно напоминающем язык человека, и отличном от машинного кода, с которым имеет дело процессор, и переводит эту программу в машинный язык, представляющий собой комбинацию нулей и единиц, которые компьютер обрабатывает как последовательности электрических импульсов. Утилиты выполняют рутинные, но часто крайне необходимые, функции, например, удаление ненужной информации с магнитных дисков. Эти скромные «рабочие лошадки» помогают решать типовые задачи обработки информации.

И так, полная *вычислительная система*, включающая аппаратуру (*hardware*) и программное обеспечение (*software*), создает среду, в которой могут разрабатываться, храниться и выполняться программы.

ПО от производителя к конечному потребителю распространяется на основании специального типа договора, который называется *лицензией*. Все лицензии на программное обеспечение, в целом, делятся на две большие категории:

1. несвободные (они же *собственнические*, *проприетарные*, полусвободные) и
2. лицензии свободного и открытого ПО.

## 2. СТРУКТУРА КОМПЬЮТЕРНОЙ ПРОГРАММЫ

Любая компьютерная программа состоит из 3-х частей (Рис. 3.2):

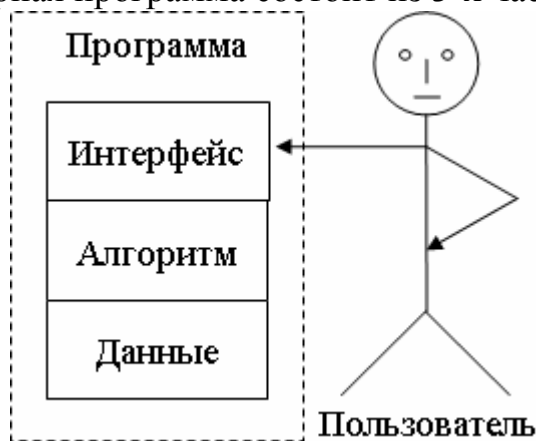


Рис. 3.2. Структура компьютерной программы

1. *Интерфейс* – набор правил взаимодействия между пользователем и программой;

2. *Алгоритм*, или *бизнес-логика* – набор правил преобразования информации из входной в выходную;

3. *Данные* – то, что обрабатывает программа.

Все три части обязательно присутствуют в любой компьютерной программе. Однако, «удельный вес» каждой из них зависит от назначения программы. Так, например, в научно-технических (инженерных) программах преобладают алгоритмы, а интерфейс и данные у них, как правило, не сложные. Экономические задачи, наоборот, обрабатывают большие объемы информации (т.е. преобладают данные), а алгоритмы и интерфейс у таких программ относительно простые. Игровые программы обладают хорошо проработанными интерфейсами и захватывающими алгоритмами, и несложными данными.

### **3. СТРУКТУРА ОПЕРАЦИОННОЙ СИСТЕМЫ**

В любой операционной системе общего назначения можно выделить 4-е части:

1. ядро,
2. файловую систему,
3. интерфейс и
4. утилиты.

Ядро – это основная, определяющая, часть ОС, которая управляет работой аппаратной части компьютера и выполнением программ. Файловая система определяет способ хранения информации на ВЗУ компьютера. Интерфейс организует взаимодействие пользователя с компьютером. И, наконец, утилиты – это просто отдельные программы, которые выполняют некоторые служебные функции, например – копирование или удаление файлов.

### **4. ПОЛИМОРФИЗМ**

*Полиморфизм (polymorphism)* – это свойство компьютерных программ, которое позволяет использовать одни и те же приемы и методы для решения нескольких схожих, но технически различных задач. В общем виде идея полиморфизма может быть коротко сформулирована следующим образом – «один интерфейс – и множество реализаций».

Принцип полиморфизма не является чем-либо совершенно новым, или «революционным изобретением» программистов, – он уже давно и широко используется в различных областях науки и техники. Так, например, последовательность расположения педалей сцепления, тормоза и газа совершенно одинаков во всех автомобилях, в не зависимости от типа их двигателя (карбюраторный, или дизельный), области применения (легковой или грузовой), класса, марки, места производства и т.д. Это позволяет водителям

совершенно легко и просто «пересаживаться» с одного автомобиля на другой. С другой стороны, не составит особого труда представить себе «коммерческий успех» автомобиля, у которого конструкторы «оптимизировали» и, соответственно, изменили (без веских на то оснований) порядок расположения этих педалей! Единственное, что сделали программисты, так это то, что они дали подходящее название одному из самых существенных признаков своего товара.

В программировании полиморфизм достаточно широко используется при создании самых различных программ, самого разнообразного назначения. При этом, однажды освоив некоторый набор приемов работы с одним приложением, затем большинство из них можно использовать и при работе с другими приложениями. Естественно, что в каждом конкретном приложении, и его состоянии, могут быть задействованы совершенно разные механизмы. Так, например, перемещение или копирование выделенного фрагмента в пределах одного документа реализуются с помощью элементарных функций перемещения и копирования конкретного приложения. Те же действия, но между различными документами одного приложения, могут задействовать совсем другие механизмы (например, технологию DDE – Dynamic Data Exchange), а между различными приложениями – третьи (например, технологию OLE – Object Linking and Embedding). Однако, с точки зрения пользователя, все эти «тонкости реализации» никакого значения не имеют – для него существенным является лишь то, что чтобы одинаковые действия в различных приложениях приводили бы к одинаковым результатам.

Следование принципам полиморфизма, как и любым стандартам, вообще, имеет как свои сильные, так и слабые стороны. Так, в науке любые стандарты являются своеобразным «тормозом» для достижения новых результатов. В промышленности же, при производстве товаров, например, следование стандартам является необходимостью, поскольку предоставляет много дополнительных преимуществ. В частности, при массовом производстве использование унифицированных технологий всегда является более дешевым. Пользователи также всегда отдадут предпочтение тем товарам, которые им давно и хорошо известны, и совсем неохотно «осваивают» совершенно новые изделия – все люди немножечко ленивы, и не любят «переучиваться».

А поскольку, программы также являются товаром, то, естественно, что большинство разработчики последовательно и целенаправленно придерживаются идеи полиморфизма. Для пользователей же эта особенность означает лишь одно – что, однажды освоив интерфейс и приемы работы с одной из программ, затем, их можно использовать и при работе с другими приложениями Microsoft Windows. Поэтому, данную методику следует освоить как можно раньше, и как можно лучше, поскольку ею придется пользоваться постоянно и повсеместно.



## ЛЕКЦИЯ №4

### СЕТЕВЫЕ ТЕХНОЛОГИИ

#### План

1. Компьютерные сети
2. Простейшая компьютерная сеть
3. Классификация компьютерных сетей
4. Топологии компьютерных сетей
5. Модель взаимодействия открытых систем ISO/OSI
6. Понятие протокола и интерфейса

### 1. КОМПЬЮТЕРНЫЕ СЕТИ

*Компьютерная сеть* – это совокупность компьютеров, между которыми возможен прямой обмен информацией без использования промежуточных носителей. Для создания компьютерной сети необходимо выполнение двух условий:

1. чтобы входящие в нее компьютеры были соединены между собой *каналами передачи информации (каналами связи)*, и
2. чтобы на каждом из них было установлено *специальное программное обеспечение для поддержания работы сети (программы управления сетью)*.

При этом технология работы в сети зависит как от способа организации каналов связи, так и от программного обеспечения.

### 2. ПРОСТЕЙШАЯ КОМПЬЮТЕРНАЯ СЕТЬ

Она образуется при соединении двух компьютеров в пределах нескольких метров при помощи специального кабеля. Этот кабель называется *нуль-модемом* и подключается к последовательным (COM) или параллельным (LPT) портам обеих компьютеров. Такое соединение компьютеров, как правило, является временным и называется *прямым компьютерным соединением*. Оно может быть установлено, а затем снято любым конечным пользователем. Для каждого вида соединения (через COM или LPT-порты) необходим свой вид кабеля. При этом в случае соединения компьютеров через параллельные (LPT) порты обмен информацией выполняется 2 – 4 раза быстрее.

 *Нуль-модемный кабель присоединяется при выключенных компьютерах!*

Прямые компьютерные соединения используются в основном для обмена информацией между портативными и стационарными офисными компьютером, хотя такой обмен возможен и между двумя стационарными компьютерами.

При прямом компьютерном соединении возможны два типа их взаимодействия:

- *прямой доступ*, при котором возможна только пересылка информации с одного компьютера на другой и
- *удаленное управление*, при котором возможно выполнение программ, размещенных на другом компьютере.

Операционная система Microsoft Windows внутренними средствами удаленного управления для прямого кабельного соединения не обладает.

В последнее время для обмена информацией между двумя или несколькими компьютерами все более широкое распространение получают *беспроводные (wireless)* способы соединения, в которых используются электромагнитные волны *инфракрасного (IrDA)* и *радиочастотного (Blue Tooth, WiFi)* диапазонов. При этом способы построения таких сетей упрощаются, а функциональные возможности – повышаются.

### 3. КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

В зависимости от компьютеров, каналов связи, установленного программного обеспечения и некоторых других признаков компьютерные сети принято классифицировать по:

- типу среды передачи данных;
- способу организации взаимодействия компьютеров;
- скорости передачи данных;
- занимаемой территории;
- ведомственной принадлежности;
- топологии.

В зависимости от **типа среды передачи данных** компьютерные сети делятся на:

- *проводные,*
- *беспроводные* и
- *смешанные* или *гибридные*, в которых одни фрагменты – проводные, а другие – беспроводные.

По **способу организации взаимодействия компьютеров** различают два типа сетей:

- *одноранговые сети* и
- *сети с выделенным сервером* (типа *клиент/сервер*).

**Одноранговые сети** не предусматривают выделение специальных компьютеров, организующих работу сети. Каждый пользователь, подключаясь к сети, выделяет в сеть какие-либо ресурсы (дисковое пространство, принтеры) и подключается к ресурсам, предоставленным в сеть другими пользователями. Такие сети просты в установке и наладивании; они существенно дешевле сетей с выделенным сервером. В свою очередь, **сети с выделенным сервером**, несмотря на сложность настройки и относительную дороговизну, позволяют осуществлять централизованное управление.

По скорости передачи данных сети делятся на:

- *низко-*,
- *средне- и*
- *высокоскоростные.*

Основной критерий разделения – скорость передачи данных. Понятно, что скорость передачи данных – понятие не постоянное. То, что сегодня считается среднескоростным соединением, завтра будет отнесено к низкоскоростным. Тем не менее, сегодня низкоскоростной сетью считается сеть со скоростью передачи информации до 10 Мбит/с. Среднескоростная сеть передает данные со скоростью до 100 Мбит/с, а высокоскоростные сети передают информацию со скоростью свыше 100 Мбит/с.

По занимаемой территории сети могут быть:

- *локальными,*
- *региональными (муниципальными) и*
- *глобальными.*

**Локальная вычислительная сеть (ЛВС)** или по-английски *Local Area Network – LAN*, объединяет компьютеры, расположенные в пределах одного помещения, внутри одного или нескольких зданий, между которыми необходимо организовать постоянный обмен информацией. Такие сети создаются при помощи специального стационарного сетевого оборудования.

**Региональная вычислительная сеть** по-английски называется *Metropolitan Area Network – MAN* и по своему функциональному назначению такая сеть эквивалентна локальной вычислительной сети. Она, как правило, объединяет в единое целое несколько сетей значительно удаленных друг от друга в пределах одного населенного пункта, или одного региона, между которыми необходимо организовать постоянный обмен большими потоками информации. Компьютеры в такой сети соединяются при помощи постоянно действующих выделенных каналов связи. При этом среда передачи данных сети MAN может быть как проводной, так и беспроводной. Сети, связывающие значительно удаленные друг от друга компьютеры, так же называются *распределенными*.

**Глобальная сеть** по-английски называется *Wide Area Network – WAN*. Она объединяет компьютеры, расположенные по всему миру, на которых имеются огромные объемы разнообразной информации и доступной всем желающим как свободно, так и на коммерческой основе. Компьютеры в такой сети связаны каналами с очень высокой пропускной способностью. Наиболее известной глобальной сетью является *Интернет*, хотя имеются и другие, например, *Microsoft on Line, America on Line*.

В зависимости от **прав собственности на сети (ведомственной принадлежности)** последние могут быть двух видов:

1. *общего пользования (public)* и

## 2. частные (private).

Среди сетей общего пользования выделяются телефонные сети (*PSTN – Public Switched Telephone Network*), сети передачи данных (*PSDN – Public Switched Data Network*) и высокоскоростные цифровые сети с предоставлением комплексных услуг (*ISDN – Integrated Services Digital Network*).

Информационные системы, в которых средства передачи данных принадлежат одной компании и используются только для нужд этой компании, принято называть *Сетью Масштаба Предприятия* или *Корпоративной Сетью* (*Enterprise Network*).

## 4. ТОПОЛОГИИ КОМПЬЮТЕРНЫХ СЕТЕЙ

При объединении в сеть более двух компьютеров всегда необходимо решать задачу, каким образом их соединить друг с другом. И при увеличении числа связываемых устройств, число возможных способов конфигураций резко возрастает. Каждый из таких вариантов физических связей образует конфигурацию или *топологию* сети.

*Под топологией сети понимается граф, вершинам которого соответствуют компьютеры или другое коммуникационное оборудование, а ребрам – каналы связи между ними.*

Среди множества конфигураций компьютерных сетей различают:

1. полносвязные и
2. неполносвязные.

*Полносвязная топология* (Рис. 4.1, а) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Это самая надежная топология сети. Но, несмотря на кажущуюся простоту, этот вариант оказывается громоздким и не эффективным потому, что каждый компьютер должен иметь достаточное количество коммуникационных портов для связи со всеми остальными компьютерами сети. Поэтому практически ни когда не используется, поскольку является самым дорогим и трудным в обслуживании. Этот вид топологии применяется лишь в многомашинных комплексах или сетях, объединяющих небольшое количество компьютеров.

Все другие варианты (Рис. 4.1, б ÷ ж) основаны на *неполносвязных топологиях*, когда для обмена данными между двумя компьютерами может потребоваться транзитная передача данных через другие узлы сети.

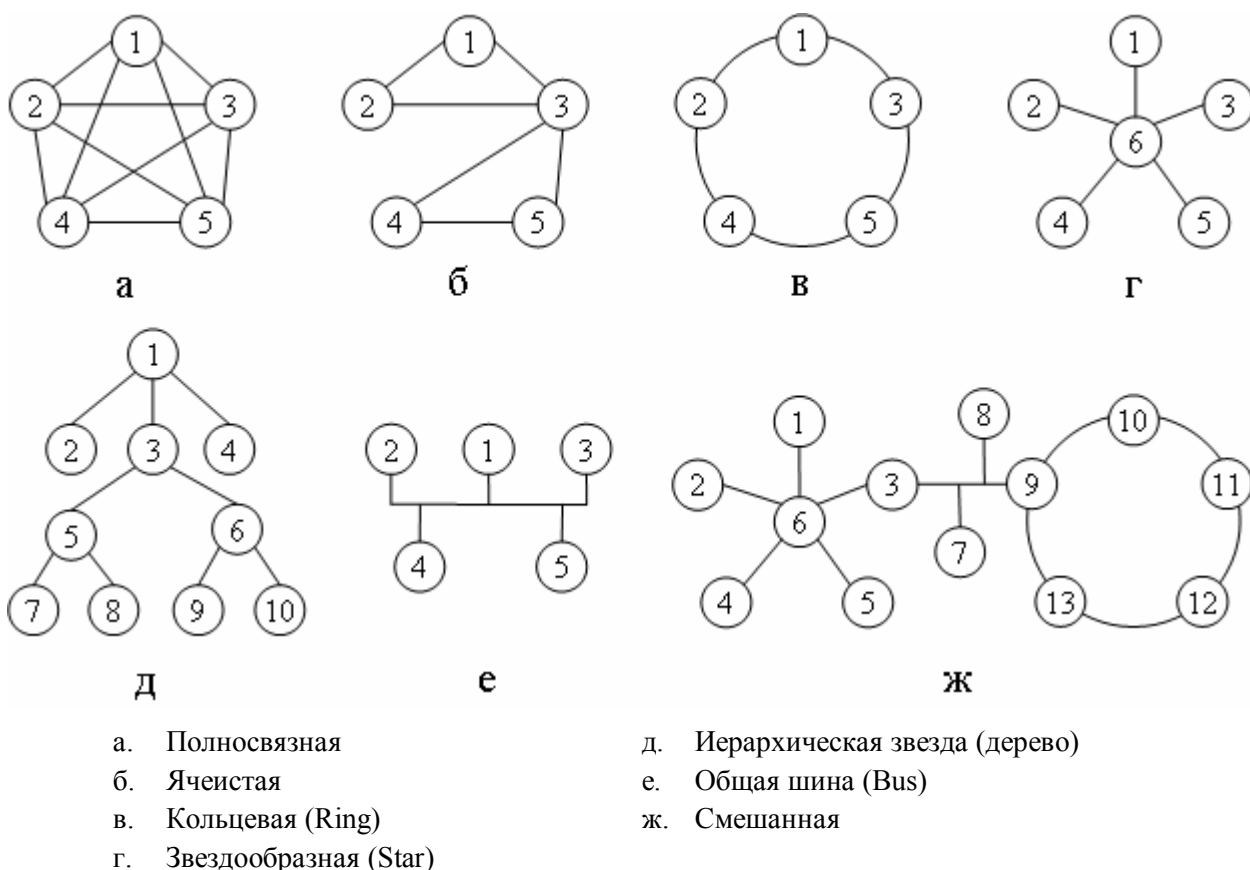


Рис. 4.1. Типовые топологии компьютерных сетей

*Ячеистая топология* (Рис. 4.1, б) получается из полностью связанной путем удаления некоторых связей. Она допускает соединения большого количества компьютеров и характерна, как правило, для крупных сетей. В такой сети есть, по крайней мере, два узла имеющих два или более пути между ними.

В сетях с *кольцевой топологией* (англ. *Ring*) данные передаются по кольцу от одного компьютера к другому (Рис. 4.1, в). Главным достоинством такой топологии является то, что она по своей природе обеспечивает резервирование связей. Действительно, любая пара узлов соединена здесь двумя путями – по часовой стрелке и против нее. Кольцо представляет собой очень удобную конфигурацию и для организации обратной связи – данные, сделав полный оборот, возвращаются к узлу источнику. Поэтому источник может контролировать процесс доставки информации потребителю. Часто это свойство кольца используется для тестирования связанной сети и поиска неисправностей. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца.

*Звездообразная топология* (англ. *Star*) образуется, когда каждый компьютер подключается непосредственно к центральному устройству, которое называется *концентратором* (Рис. 4.1, г). В его функции входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В

качестве концентратора может выступать как универсальный компьютер, так и любое специализированное многоходовое устройство, такое, например, как коммутатор (англ. *switch*), маршрутизатор (англ. *router*) или тот же физический концентратор (англ. *hub*). К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически связанных между собой связями типа звезда (Рис. 4.1, д). Полученную в результате структуру называют *иерархической звездой* или *деревом*. В настоящее время дерево является самой распространенной топологией связей, как в локальных, так и в глобальных сетях.

Конфигурация *общая шина* (англ. *Bus*) является особым частным случаем звезды (Рис. 4.1, е). В такой сети все узлы подключены к единой среде передачи данных, например, к коаксиальному кабелю. Аналогичную топологию имеют многие сети, использующие беспроводную связь, в которых в качестве общей шины выступает общий радиоканал. Передаваемая информация распространяется по шине и одновременно доступна всем компьютерам, присоединенным к ней. Основными преимуществами такой схемы являются дешевизна и простота присоединения новых узлов к сети, а недостатками – низкая надежность (любой дефект шины полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится между всеми узлами сети).

В то время как небольшие сети, как правило, имеют типовую топологию такую, например, как звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить лишь отдельные фрагменты (подсети), имеющие типовую топологию. Поэтому такие сети называются сетями со *смешанной топологией* (Рис. 4.1, ж).

## **5. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ ISO/OSI**

К концу 70-х годов XX столетия в мире уже существовало большое количество всевозможных компьютерных сетей, построенных по самым разнообразным технологиям. Такое разнообразие сетевых технологий вывело на первый план проблему обмена информацией между ними. Решение этой проблемы относится к области стандартизации. Оно было получено в 80-е годы, когда Международная организация по стандартизации (*ISO – International Standards Organization*) разработала, а сообщество сетевых проектировщиков приняло, стандартную *модель взаимодействия открытых систем* (*OSI – Open System Interconnection*).

Основная идея модели OSI заключается в том, что общая задача передачи данных расчленяется на отдельные, легкообозримые части (или уровни). Она разрабатывалась в качестве своего рода универсального языка сетевых спецификаций и сыграла значительную роль в развитии компьютерных сетей. Именно поэтому ее еще называют *справочной моделью, семиуровневой моделью OSI* или, просто, *моделью OSI*. Не смотря на то, что модель OSI сугубо теоретическая, на практике она оказалась исключительно полезной – без нее в сетях воцарился бы полный хаос, когда оборудование от разных производителей не смогло бы работать вместе. Модель OSI позволяет, в случае необходимости, изменить или заменить какой-либо отдельный элемент или составляющую сети без изменения всей сети. При этом также не требуется перепроектирование всей сети с нуля.

*Модель OSI определяет:*

1. уровни взаимодействия в компьютерных сетях,
2. стандартные названия уровней и
3. функции, которые должен выполнять каждый уровень.

*Модель OSI не содержит описаний реализаций конкретного набора правил сетевого взаимодействия.*

В модели OSI (Рис. 4.2) средства сетевого взаимодействия делятся на семь уровней:

1. физический,
2. канальный,
3. сетевой,
4. транспортный,
5. сеансовый,
6. представления и
7. прикладной.

Каждый из них имеет дело с совершенно определенным аспектом сетевых взаимодействия.

Пусть, например, приложению **А**, выполняющемуся на компьютере №1, необходимо переслать информацию приложению **Б** на компьютер №2. Для этого приложение **А** через интерфейс прикладного программирования (*API – Application Program Interface*) обращается с запросом к прикладному уровню. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. После формирования сообщения прикладной уровень направляет его вниз на уровень представления. После выполнения необходимых действий на этом уровне сообщение передается на следующий уровень вниз и т. д., до достижения физического уровня.

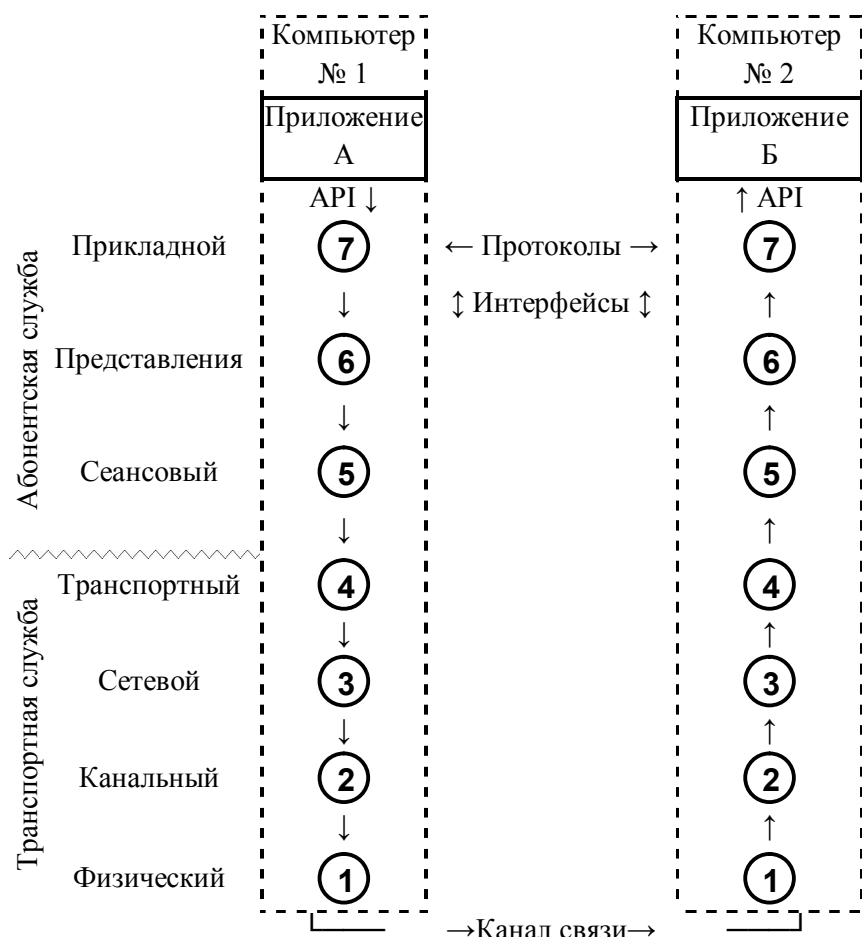


Рис. 4.2. Модель взаимодействия открытых систем ISO/OSI

На физическом уровне сообщение передается на выходной интерфейс компьютера №1, и далее оно начинает свое «путешествие» по сети. До этого сообщение передавалось от одного уровня к другому в пределах компьютера №1. Когда сообщение по сети поступает на входной интерфейс компьютера №2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает сообщение, выполняя при этом все необходимые действия, и передает его вышележащему уровню. В результате информация, переданная приложением А из компьютера №1, будет получена приложением Б на компьютере №2.

Рассмотренная схема взаимодействия приложений соответствует идеальной модели OSI, когда прикладная программа обращается с запросом к самому верхнему уровню – прикладному. Однако на практике во многих случаях имеется возможность прикладным программам напрямую обращаться к нижележащим уровням. Например, некоторые Системы Управления Базами Данных (СУБД) имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленной информации, не использует средства файловой системы. Оно обходит верхние уровни модели OSI и непосредственно обращается к ответственным за транспортировку средствам, которые располагаются на более низких уровнях модели OSI.



**Физический уровень** (англ. *physical layer*) имеет дело с передачей потока битов по физическим каналам связи, таким, например, как кабели или беспроводные каналы передачи информации. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или одним из портов. Физический уровень не вникает в смысл информации, которую он передает. Для него эта информация представляется однородным потоком битов, которую нужно доставить адресату без искажений.


**Канальный уровень** (англ. *data link layer*). В некоторых сетях, в которых линии связи используются попеременно (разделяются) несколькими парами взаимодействующих компьютеров, физическая среда передачи данных может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и исправления ошибок. Для этого на канальном уровне биты группируются в наборы, называемые *кадрами (frames)*. Канальный уровень обеспечивает корректность передачи кадров, помещая в его начало и конец специальную последовательность бит – контрольную сумму.

Функции средств канального уровня определяются по-разному для локальных и глобальных сетей:

- В *локальных сетях* канальный уровень должен обеспечить доставку кадра между любыми узлами сети.
- В *глобальных сетях* канальный уровень должен обеспечить доставку кадра только между двумя *соседними* узлами.

Топология сети (шина, звезда и т. д.) определяется как раз на канальном уровне. На этом уровне также определяется работа с так называемыми МАС-адресами. *МАС-адрес* – это уникальный аппаратный адрес сетевого устройства, например, сетевого адаптера, точки доступа и т. д. Каждому производителю сетевых устройств выделяется свой диапазон МАС-адресов. В мире не существует двух сетевых устройств с одинаковыми МАС-адресами.

**Сетевой уровень** (англ. *network layer*) служит для образования единой транспортной системы, объединяющей несколько сетей. Причем эти сети могут использовать совершенно различные принципы передачи сообщений и обладать произвольной структурой.

 **Технология, позволяющая соединить в единую сеть множество сетей, в общем случае построенных на основе разных технологий, называется технологией межсетевого взаимодействия (*internetworking*).**

Функции сетевого уровня реализуются:

1. группой протоколов и
2. специальными устройствами – *маршрутизаторами* (англ. *routers*).

Маршрутизатор – это устройство, которое соединяет между собой различные сети. Он собирает информацию о топологии межсетевых соединений и на ее основе пересылает сообщения. Такие сообщения на сетевом уровне называются *пакетами* (англ. *packets*). Для того чтобы передать пакет от отправителя,

находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество *транзитных передач между сетями*. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет. Единица такого перехода называется *хопом* (от англ. *hop* – прыжок). Количество хопов равно количеству маршрутизаторов между двумя сетями. Проблема выбора наилучшего пути называется *маршрутизацией*, и ее решение является одной из главных задач сетевого уровня.

Маршрутизатор может быть реализован программно, на базе универсального компьютера, работающего под управлением операционной системы UNIX или Windows. Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня.

Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол обмена пакетами IPX стека Novell.

*Транспортный уровень* (англ. *transport layer*) обеспечивает верхним уровням модели OSI – прикладному, представлению и сеансовому – передачу данных с той степенью надежности, которая им требуется. То есть он отвечает за обнаружение и исправление таких ошибок, как искажение, потеря и дублирование пакетов.

Выбор глубины и качества обнаружения и исправления ошибок передачи данных определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких уровней. С другой стороны этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая более низкими уровнями – сетевым, канальным и физическим. Так, если качество каналов связи высоко и вероятность возникновения ошибок невелика, то разумно воспользоваться одним из облегченных вариантов. Если же транспортные средства нижних уровней не очень надежны, то целесообразно использовать максимум средств обнаружения и устранения ошибок.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети – компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протокол TCP стека TCP/IP или SPX стека Novell.

*Сеансовый уровень* (англ. *session layer*) обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставлять средства синхронизации сеанса. Эти средства синхронизации позволяют в ходе длительных передач данных сохранять информацию о состоянии этих передач в виде контрольных точек, чтобы в случае отказа можно было вернуться назад к последней из них, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов. Функции этого уровня часто объединяются с функциями прикладного уровня и реализуются в одном протоколе.

*Уровень представления* (англ. *presentation layer*), как явствует из его названия, имеет дело с формой представления передаваемой по сети

информации, не меняя при этом ее содержание. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолевать синтаксические различия в представлении данных или же различия в их кодировках. На этом уровне может выполняться шифрование и дешифрование информации, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является *слой защищенных сокетов (SSL – Secure Socket Layer)*, который обеспечивает секретность обмена сообщениями для протоколов прикладного уровня стека TCP/IP.

*Прикладной уровень* (англ. *application layer*) – это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперируют на прикладном уровне, обычно называется *сообщением (message)*.

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. Например, наиболее распространенные реализации сетевых файловых служб являются NCP в операционной системе Novell NetWare, SMB в Microsoft Windows, NFS в стеке TCP/IP. На этом уровне работает множество самых разнообразных протоколов, например, HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol) и т. д.

Четыре нижних уровня стандартной модели OSI образуют *транспортную службу* компьютерной сети, которая полностью решает задачу транспортировки сообщений с заданным уровнем качества, освобождая более высокие уровни от решения этих задач. В свою очередь, три верхних уровня, обеспечивающие логическое взаимодействие прикладных процессов, функционально объединяются в *абонентскую службу*.

## 6. ПОНЯТИЕ ПРОТОКОЛА И ИНТЕРФЕЙСА

В сущности, термин «протокол» и «интерфейс» выражают одно и то же понятие – формализованное описание процедуры взаимодействия двух объектов. Но традиционно в сетях за ними закрепились разные области действий. *Протоколы* определяют правила взаимодействия модулей одного уровня в разных узлах (*по горизонтали*), а *интерфейсы* – правила взаимодействия модулей соседних уровней в одном узле (*по вертикали*).

## ЛЕКЦИЯ №5

### ВСЕМИРНАЯ КОМПЬЮТЕРНАЯ СЕТЬ ИНТЕРНЕТ

#### План

1. Краткая история создания Интернета
2. Структура и принципы работы Интернета
3. Адресация компьютеров в сети
4. Доменная система имен
5. Единообразный локатор ресурса

### 1. КРАТКАЯ ИСТОРИЯ СОЗДАНИЯ ИНТЕРНЕТА

Непосредственной предшественницей Интернета была компьютерная сеть *ARPANET*, созданная в 1969 году рядом научных и исследовательских организаций под контролем одного из подразделений Министерства обороны США – агентства *ARPA* (Advanced Research Projects Agency – Агентство Передовых Исследовательских Проектов). В разные годы своего существования агентство несколько раз меняло название. Так, с момента создания в 1958 г. оно называлось *ARPA*; затем в 1972 г. было переименовано в *DARPA* (с добавлением слова *Defense* – Оборона); затем в 1993 – опять в *ARPA*, и, наконец, 11 марта 1996 года – снова в *DARPA*. Основная цель, которая ставилась при создании сети, состояла в том, что она должна выполнять свои функции даже при частичном ее разрушении. Например, в результате военных действий, пожара, стихийного бедствия или простого воровства оборудования. Это был *эксперимент по пакетной коммутации сообщений*.

Первоначально сеть *ARPANET* состояла из 4-х узлов (Рис. 5.1):

1. компьютер *SDS SIGMA* в Калифорнийском университете Лос-Анджелеса, *UCLA* (внизу),
2. компьютер *SDS940* в Стэнфордском исследовательском институте, *SRI* (вверху в центре),
3. компьютер *IBM360* в Калифорнийском университете в городе Санта-Барбара, *UCSB* (слева),
4. компьютер *DEC PDP-10* в университет штата Юта в Солт-Лейк Сити, *UTAH* (справа).

В сети *ARPANET* все узлы (компьютеры) имеют одинаковый статус. При этом каждый из них уполномочен порождать, передавать и получать сообщения от любого другого. Сообщения для передачи разбиваются на небольшие стандартизированные элементы, называемые *пакетами*. Каждый такой пакет имеет адрес назначения, и доставка сообщения обеспечивается тем, что каждый узел имеет возможность посылать (или переадресовывать) пакеты по сети к месту назначения. Компьютеры в сеть объединялись при помощи

разработанных в BBN – Bolt, Beranek & Newman (Кембридж, шт. Массачусетс), устройств *IMP* (*Interface Message Processor* – *Интерфейса Процессора Сообщений*), размером с холодильник, которые осуществляли связи между компьютерами через *телефонную сеть*.

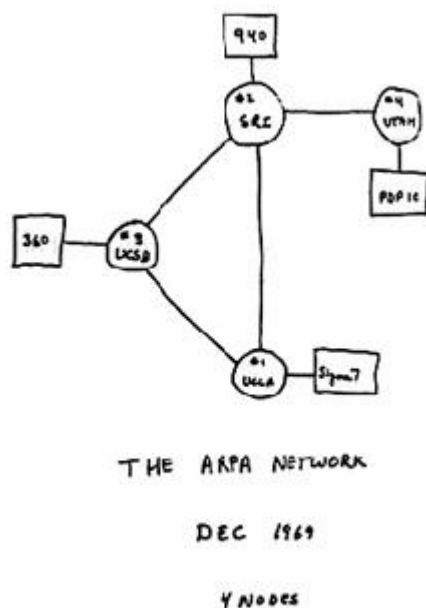


Рис. 5.1. Первоначальный вариант сети ARPANET

Первые испытания ARPANET оказались крайне успешными. Ученые исследовательских учреждений, послуживших испытательными полигонами, получили возможность передавать данные и совместно пользоваться удаленным доступом к компьютерам. К лету 1970-го года сеть насчитывала уже целых 10 узлов, а через год их стало уже 15. Университет Гавайи обзавелся собственной сетью, напоминающей ARPANET – *ALONANET*. С каждым годом ARPANET росла и развивалась – в нее включались все новые и новые участники. Право доступа в сеть начали требовать себе сначала все крупные лаборатории, потом – и более мелкие. Наконец, в гонку за ARPANET включились и высшие учебные заведения. Военные ворчали, но соглашались. В 1972 году сеть уже связывала 50 университетов и исследовательских центров.

Одновременно с ростом сети велись работы по созданию функционально полного протокола межкомпьютерного взаимодействия и другого сетевого программного обеспечения. В декабре 1970 года Сетевая рабочая группа (Network Working Group – NWG) под руководством С. Крокера завершила работу над первой версией протокола, получившего название *Протокол управления сетью* (*Network Control Protocol* – *NCP*). После того, как в 1971 – 1972 годах были выполнены работы по реализации NCP на узлах ARPANET, пользователи сети, наконец, смогли приступить к разработке приложений.

В 1972 году появилось первое «сетевое» приложение – *электронная почта*. В марте Рэй Томлинсон (Ray Tomlinson) из BBN, движимый

необходимостью создания для разработчиков ARPANET простых средств координации, написал базовые программы пересылки и чтения электронных сообщений. Он же предложил использовать значок «@», который и по сей день является неотъемлемой частью любого электронного адреса. Интересно, что в разных странах его называют совершенно по-разному: у нас – это «собачка», в Дании – «хобот слона», в Греции – «маленькая утка», а в Германии – «висящая обезьяна». Чего-чего, а чувства юмора программистам всего мира занимать не приходится. В июле того же года Лоуренс Робертс добавил к этим программам возможности выдачи списка сообщений, выборочного чтения, сохранения в файле, пересылки и подготовки ответа. Совместимость различных почтовых систем, работающих на разных компьютерных платформах, продемонстрировала выгоды массовых электронных коммуникаций между людьми. С тех пор более чем на десять лет электронная почта стала крупнейшим сетевым приложением. Для своего времени электронная почта стала тем же, чем в наши дни является Всемирная Паутина – исключительно мощным катализатором роста всех видов межперсональных потоков данных.

В октябре 1972 года решено было провести Международную конференцию по компьютерным коммуникациям (International Computer Communication Conference – ICC3). Идея была в том, чтобы установить временный интерфейсный процессор (TIP – Terminal Interface Processor) в здании Washington Hilton Hotel и дать публике войти и использовать ARPANET, выполняя приложения по всем Штатам. Готовились почти год. В подготовку к демонстрации вовлекли лучших специалистов из ARPA и BBN. Боб Кан (Bob Kahn) из BBN в течение трех дней весьма успешно демонстрировал работу сети. Профессионалы не хотели, а может быть, и не могли поверить, что сеть, содержащая 100 компьютеров, может надежно работать. Это был первый показ на публике новой сетевой технологии.

После конференции наступает новый этап в развитии ARPANET. В 1973 году впервые были подключены зарубежные узлы – Университетский колледж в Лондоне (University College of London) и Королевская лаборатория радиолокации в Норвегии (Rogee Radar Establishment). Тогда же была запущена спутниковая линия связи с Гавайским университетом. С этого момента ARPANET становится международной и основной задачей становится объединение разнородных сетей в единую сеть.

В середине 1975 года DARPA, пришло к выводу, что ее сеть стабильна, и управление ARPANET было передано DCA – Defense Communications Agency – Оборонному Агентству по Коммуникациям США (в настоящее время оно называется DISA – Defense Information Systems Agency – Оборонное Агентство по

Информационным Системам). Так ARPANET превратилась из экспериментальной сети в рабочую сеть.

Тем не менее, DARPA (и не только оно) продолжало заниматься техническими аспектами ARPANET. Одним из таких направлений была разработка межсетевых протоколов, так как РР-протоколы (Point-to-Point – протоколы типа Точка-Точка) уже не могли обеспечивать подключения такого большого количества различных по структуре локальных сетей, желающих подключиться к ARPANET. Это было обусловлено спецификой технической реализации сети, которая состояла в том, что основой ARPANET являлись соединенные между собой IMP-узлы. С течением времени IMP-узлы были переименованы в PSN-узлы (*Packet Switch Node – Узлы Коммутации Пакетов*) и само оборудование, представляющее собой эти устройства IMP/PSN, было модернизировано. PSN-узлы были связаны между собой каналами связи типа точка-точка. Причем, связаны они так, чтобы каждый PSN-узел имел, как минимум, два канала связи с двумя разными PSN-узлами. При таком положении, если откажет один канал связи или один PSN-узел, связь в сети ARPANET не будет нарушена, так как другие PSN-узлы смогут отправить свои пакеты в обход аварийного участка. Каждый PSN-узел укомплектован двадцатью двумя внешними портами, к которым можно подключать клиентские машины. Машина, подключенная к порту PSN-узла, называется *хостом*. Обмен данными между хостом и PSN-узлом происходил по протоколу X.25. Для идентификации машины в сети ARPANET использовалась следующая схема адресации: каждый PSN-узел получает свой уникальный номер, а так как каждый порт PSN-узла тоже имеет конкретный номер, получается, что адрес конечного получателя – хост-машины – состоит из двух чисел: номера PSN-узла и номера порта, к которому подключена эта хост-машина.

Конец 70-ых и начало 80-ых годов XX столетия характеризуются бурным ростом компьютерных сетей, которые были построены по совершенно разным технологиям и предназначены для решения совершенно различных задач. При этом многие понимали, что от возможности обмена информацией между сетями все только бы выиграли. Однако все эти сети использовали совершенно разные протоколы, несовместимые между собой. То есть, возникла необходимость разработки нового протокола, который бы, не нарушая структуры существующих сетей, позволял им обмениваться информацией, и работал бы на любой аппаратной платформе. Среди множества этих сетей особенно выделялась ARPANET. Ее децентрализованная структура, существенно отличающаяся от структур, существовавших в то время корпоративных сетей, позволяла подключать к сети компьютеры практически любого типа, при одном лишь условии – что бы они «понимали» протокол

пакетной передачи данных NCP. Но существующий вариант протокола NCP имел ограничения, не позволяющие подключать к ARPANET такое большое количество сетей и компьютеров. В сентябре 1973 года Винт Серф и Боб Кан, входившие в Международную сетевую рабочую группу (INWG), распространили на специальной встрече этой группы, состоявшейся во время конференции в Университете Суссекса, первую документированную версию выработанных ими спецификаций – *семейства протоколов TCP/IP*. В июле 1977 года Серф и Кан впервые продемонстрировали передачу данных с использованием TCP по трем различным сетям. Пакет прошел по следующему маршруту: Сан-Франциско – Лондон – Университет Южной Калифорнии. В конце своего путешествия пакет проделал 150 тысяч км, не потеряв при этом ни одного бита! А 1 января 1983 года ARPANET перешла с протокола NCP на TCP/IP. Это был переход в стиле «Дня X», требующий одновременных изменений на всех компьютерах. Переход тщательно планировался всеми заинтересованными сторонами в течение нескольких предшествующих лет и прошел на удивление гладко.

Внедрение протокола TCP/IP, позволявшего пользователям с легкостью подключаться к сети при помощи обычной телефонной линии, совпало с другим событием – разделением ARPANET. Терпению военных пришел конец – их родная, лелеемая и подкармливаемая серьезными капиталовложениями сеть, перестала обеспечивать необходимый уровень секретности. Поэтому она была разделена на две сети. Первая из них – *Defense Data Network (DDN)* – Оборонная Сеть Передачи Данных, известная так же как *MILNET – Military Network* – сеть военных – стала использоваться исключительно для военных целей. Вторая же, ARPANET, по-прежнему использовалась для исследовательских целей. Таким образом, семейство протоколов TCP/IP позволило не только объединять сети, но и разъединять их. Термин «Интернет» стал использоваться для обозначения единой сети: MILNET плюс ARPANET. В 1989 году обслуживание ARPANET полностью прекратилось, и в 1991 году она закончила свое существование. Ее функции продолжила *NSFNET* (National Science Foundation Network), объединившая к тому времени большинство сетей и практически ставшая Интернетом в том виде, в котором он сейчас есть.

Официально NSFNET была сформирована в 1986 году одним из подразделений Национального Научного Фонда США (NSF – National Science Foundation) – Отделом Сетевых и Коммуникационных Исследований и Инфраструктуры. NSFNET была составлена из более мелких сетей (включая известные тогда сети Usenet и Bitnet) и имела гораздо большую пропускную способность, чем ARPANET. Она основывалась на протоколе TCP/IP, имела иерархическую структуру и концентрировалась вокруг крупных



университетских центров США. Чтобы университет мог получить от NSF средства на подключение к Интернету, он, как было записано в программе NSFNET, «должен обеспечить доступность этого подключения для *всех* подготовленных пользователей в университетском городке».

Сеть NSFNET – это пять *суперкомпьютерных центров*, обычно называемые «магистральным хребтом Интернет в США» (*Internet Backbone*), которые были соединены специальными телефонными линиями с пропускной способностью 56 Кбит/сек. Сеть NSFNET представляла собой крупную опорную сеть, объединяющую вокруг себя более мелкие сети. Любой, кто мог установить связь с сетью NSFNET, получал доступ ко всем объединенным вокруг нее сетям. Она, по сути, была зеркальным отражением ARPANET. Некоторое время они работали параллельно. Региональные сети на базе протокола TCP/IP были соединены друг с другом через NSFNET, которая имела связь с ARPANET. Подключения осуществлялись главным образом через NSFNET, так как она обладала более высокой скоростью передачи данных, к ней легче было подсоединиться и обходилось это дешевле. Название «Интернет» начало плавно переходить от ARPANET к NSFNET.

Совместное использование суперкомпьютерных центров позволяло использовать и множество других вещей, не относящихся к суперкомпьютерам. Неожиданно университеты, школы и другие организации осознали, что заимели под рукой море данных и целый мир пользователей. Поток сообщений в сети (*трафик*) нарастал все быстрее и быстрее пока, в конце концов, не перегрузил управляющие сетью компьютеры и связывающие их телефонные линии. В 1987 году NSF заключил с компаниями Merit Network Inc., IBM и MCI контракт на управление и развитие сети. Последний предполагал замену линий связи, ранее использовавшихся в NSFNET, на магистрали T1 с пропускной способностью 1.544 Мбит/сек, а также объединение шести региональных сетей, пяти существующих суперкомпьютерных центров, сети MERIT и Национального центра атмосферных исследований в единую сеть. Эта работа была завершена в июле 1988 года. Сетевые управляющие машины также были заменены на более совершенные.

Процесс совершенствования и модернизации сети идет непрерывно. Однако большинство этих перестроек происходит незаметно для пользователей. Включив компьютер, вы не увидите объявления о том, что ближайшее время Интернет не будет доступен из-за его модернизации. Возможно, даже более важно то, что изменение сети и ее усовершенствование создали зрелую и практичную технологию. Проблемы были решены, а идеи развития проверены в деле. Так, в 1991 году опять-таки из-за нехватки пропускной способности NSFNET, ее магистральные каналы были

модернизированы до «статуса Т3», что соответствует скорости передачи данных 44.738 Мбит/сек.

В 1993 году NSF решил, что не может больше поддерживать быстро растущую систему. Он предложил независимым фирмам контракты на осуществление контроля над расширением Интернета. Компании откликнулись, и ответственность за развитие Интернета была передана множеству различных фирм. Сама же NSFNET была заменена *NAP – Network Access Point* – Точки Доступа Сети, которые были разбросаны по всей территории США. Через них осуществлялось взаимодействие и обмен информацией между компаниями, которые строили свои собственные опорные сети и предоставляли доступ к Интернету конечным пользователям. Количество таких компаний очень быстро росло, и они стали называться *ISP – Internet Service Provider* – Провайдер Доступа Интернет, или просто – Провайдер. Первым коммерческим ISP был The World. Контроль и финансирование NAP – высшего уровня иерархии Интернет – остался в NSF. Отдельные провайдеры частных опорных сетей образуют сложную систему связей – *Точек присутствия – Points of Presence – POP*, через которые отдельные пользователи или компании подключаются к Интернету.

NAP начала свою работу в 1995 году, приняв на себя все потоки информации NSFNET. Она имела сверхвысокую пропускную способность – до 155 Мбит/с и состояла из 4-х узлов, расположенных в Вашингтоне, Нью-Йорке, Сан-Франциско и Чикаго. 30 апреля 1995 года опорная сеть NSFNET прекратила функционировать в качестве магистрали – появился Интернет в том виде, который мы знаем сегодня.

С момента создания самого первого варианта ARPANET пользователями сети Интернет были только специалисты в области обработки информации. Однако, пересылка файлов и сообщений электронной почты – это не совсем то, что нужно большинству рядовых пользователей. Наиболее интенсивное проникновение Интернет буквально во все области человеческой деятельности приходится на 90-е годы XX столетия. И связано это было в первую очередь с созданием *Всемирной Паутины – World Wide Web – WWW*. Концепция Всемирной Паутины, в отличие от существующих к тому времени служб Интернет, таких, например, как электронная почта, дала возможность представлять информацию в естественной для человека форме – т. е. с текстом, изображениями, звуком, видео и прочими атрибутами. Документы, выполненные в этом формате, получили название *Web-страниц*. Фактически WWW – это распределенная система, гипертекстовых документов. После этого работа в Интернете перестала быть уделом профессионалов – он превратился в распределенную по миллионам

серверов единую информационную базу, навигация в которой не сложнее, чем просмотр обычной мультимедийной энциклопедии.

Создание Всемирной Паутины связано с именем Тима Бернерс-Ли (Tim Berners Lee) – сотрудника Европейского центра ядерных исследований (CERN). Он заложил три, из четырех ныне существующих, краеугольных камней WWW:

1. язык гипертекстовой разметки документов *HTML – HyperText Markup Language*,

2. протокол обмена гипертекстовой информацией *HTTP – HyperText Transfer Protocol*,

3. универсальный способ адресации ресурсов в сети *URL – Universal Resource Locator*,

4. общий интерфейс шлюзов *CGI – Common Gateway Interface*, который был добавлен позже командой из Национального Центра Суперкомпьютерных Приложений США (National Center for Supercomputer Applications – NCSA).

Первый текстовый *браузер (обозреватель* – программа для просмотра Web-страниц), созданный Тимом Бернерс-Ли в 1990 году, особой популярности не имел. Более удачной была версия браузера *Mosaic* для операционной системы UNIX, разработанная в NCSA группой программистов во главе с Марком Андресеном (Marc Andreessen) в 1993 года. С 1994 года, после выхода версий браузера *Mosaic* для операционных систем Windows и Macintosh, а вскоре вслед за этим и браузеров *Netscape Navigator* и *Microsoft Internet Explorer*, берет начало взрывообразное распространение популярности WWW, и как следствие Интернета, среди широких масс сначала в США, а затем и по всему миру. А с 1996 года Всемирная Паутина почти полностью подменяет собой понятие «Интернет».

Подводя итог краткой истории создания Интернета, можно выделить три основных этапа ее развития:

1-й этап 1969 – 1983 годы – создание первого варианта сети ARPANET, способной объединять другие сети и различные компьютеры;

2-й этап 1983 – 1990 г.г. – замена протокола NCP на TCP/IP и создание сети NSFNET;

3-й этап 1990 г. – создание Всемирной паутины.

Основные этапы и события истории создания Интернет представлены в Таблице 5.1.

**Таблица 5.1 Основные этапы создания Интернет**

Этап	Год	Событие
I	1969	Создан первоначальный вариант сети ARPANET, состоящий из 4-х компьютеров.
	1970	Сеть ARPANET объединяет 10 компьютеров. Создана первая версия протокола NCP.
	1971	Сеть ARPANET объединяет 15 компьютеров.
	1972	Создано первое «сетевое» приложение – электронная почта. Демонстрация ARPANET на Международной конференции по компьютерным коммуникациям.
	1973	Впервые к ARPANET подключены зарубежные узлы. Выработана 1-я документированная версия семейства протоколов TCP/IP.
	1975	ARPANET превратилась из экспериментальной сети в рабочую.
	1977	Демонстрация передачи данных с использованием TCP/IP по трем различным сетям.
II	1983	Переход ARPANET с протокола NCP на TCP/IP. Из ARPANET выделилась MILNET. Появился термин Интернет.
	1984	Введена система доменных имен – DNS – Domain Name System.
	1986	Создана NSFNET.
III	1990	Создана Всемирная паутина – WWW – World Wide Web.
	1991	ARPANET прекратила свое существование.
	1995	Создана NAP. Закрыта опорная сеть NSFNET.

24 октября 1995 года Федеральный Сетевой Совет (FNC) единодушно одобрил резолюцию, определяющую термин «Интернет». Это определение разрабатывалось при участии специалистов в области сетей и в области прав на интеллектуальную собственность.

**РЕЗОЛЮЦИЯ:** Федеральный Сетевой Совет признает, что следующие словосочетания отражают наше определение термина «Интернет». Интернет – это глобальная информационная система, которая:

1. логически взаимосвязана пространством глобальных уникальных адресов, основанных на Интернет-протоколе (IP) или на последующих расширениях или преемниках IP;

2. способна поддерживать коммуникации с использованием семейства Протокола управления передачей/Интернет-протокола (TCP/IP) или его последующих расширений/преемников и/или других IP-совместимых протоколов;

3. обеспечивает, использует или делает доступной, на общественной или частной основе, высокоуровневые сервисы, надстроенные над описанной здесь коммуникационной и иной связанной с ней инфраструктурой.

Винтон Серф, президент Сообщества Интернета, дает более краткое определение Интернета так: «Интернет – это глобальная сеть сетей, взаимосвязанных протоколами TCP/IP и другими коммуникационными протоколами».

## 2. СТРУКТУРА И ПРИНЦИПЫ РАБОТЫ ИНТЕРНЕТА

Интернет основана на идее существования множества независимых сетей почти произвольной архитектуры. Сам же термин «Интернет» означает – «Междусетье», сообщество сетей разного масштаба. То есть, единой сети фактически нет, а существует множество отдельных сетей, самого различного назначения и структуры, объединенных в единое целое. Одни из таких сетей могут объединять компьютеры в Вашем доме, а другие – принадлежать крупным корпорациям и соединяют компьютеры, расположенные на разных континентах. Отдельные сети в составе Интернет относительно независимы и могут развиваться по своим собственным законам и правилам, оставаясь в то же время частью единой структуры. Интернет не является единым целым и никому не принадлежит, но при этом более мелкие сети, подключенные к ней, обслуживаются отдельными организациями – *провайдерами*, являющимися собственниками «своего» участка Сети и получающими плату за предоставление доступа к ней.

Сейчас в Интернет используются практически все известные линии связи от низкоскоростных телефонных линий до высокоскоростных цифровых спутниковых каналов. Операционные системы, используемые в Интернет, также отличаются разнообразием. Большинство компьютеров сети работают под управлением операционной системы UNIX или Microsoft Windows.

Основное, что отличает Интернет от других сетей – это ее протоколы – TCP/IP. Свое название протокол TCP/IP получил от двух протоколов *TCP – Transmission Control Protocol – Протокол Управления Передачей* и *IP – Internet Protocol – Межсетевой Протокол*. Первый из них регламентирует способы передачи информации, а второй – способы адресации. Несмотря на то, что в Интернет используется большое число других протоколов, ее часто называют TCP/IP-сетью, так как эти два протокола, безусловно, являются важнейшими.

Когда мы пытаемся представить себе, что же такое Интернет и как она работает, то вполне естественно у большинства из нас возникают ассоциации с телефонной сетью. В конце концов, обе эти структуры используют электронные средства передачи, обе позволяют устанавливать соединение и передавать информацию; кроме того, в Интернет очень часто используются телефонные линии. К сожалению, это неверное представление, и оно является причиной непонимания принципов работы Интернет. Телефонная сеть – это *сеть с коммутацией каналов*. Когда производится вызов, абоненту выделяется некоторая часть этой сети. Даже когда абонент не использует ее (например, пошел позвать другого человека), она остается недоступной для других абонентов, которым в этот момент нужно позвонить. Это приводит к тому, что такой дорогой ресурс, как сеть, используется неэффективно.

Более соответствующая действительному положению вещей модель Интернета – это почтовое ведомство. Оно представляет собой *сеть с коммутацией пакетов*. Здесь нет выделенного участка сети. Почтовое отделение не будет бронировать самолет, чтобы доставить письмо по назначению – оно смешивается с другими письмами, сортируется и лишь затем – пересылается. Несмотря на то, что технологии абсолютно разные, служба доставки почты представляет собой удивительно точный аналог сети Интернет. Этой моделью будем продолжать пользоваться и далее.

Например, для того, что бы письмо дошло до адресата его необходимо вложить в конверт, написать на конверте адрес и бросить в почтовый ящик. Далее оно поступает в почтовое отделение, сортируется и вместе с другой корреспонденцией при помощи различных транспортных средств, таких как автомобили, самолеты, поезда и т. д., доставляется в другое почтовое отделение, более близкое к месту назначения. В нем письмо подвергается такой же процедуре с целью перемещения к еще более близкому, к месту назначения, почтовому отделению. И так далее, до тех пор, пока, в конце концов, письмо не попадет в нужное почтовое отделение, откуда оно будет доставлено адресату. При этом все почтовые отделения располагают информацией:

1. о своих связях с другими почтовыми отделениями, и
2. о том, какой из маршрутов будет наилучшим для перемещения корреспонденции ближе к пункту назначения.

Аналогично почтовому ведомству работает и компьютерная сеть Интернет. Только вместо писем в ней перемещаются блоки информации, которые называются *пакетами*. Каждый такой пакет помещается в специальный «конверт» – *IP-адрес*, который соответствует нужному компьютеру в сети. Аналогами почтовых отделений являются *маршрутизаторы*. Они способны по IP-адресу принятого пакета автоматически определить, на какой из соседних маршрутизаторов его необходимо переправить. Маршрутизатором может быть программа, но может быть и отдельный, специально выделенный для этой цели компьютер. Наиболее широко в Интернет используются маршрутизаторы типа NetBlazer или Cisco, чья операционная система напоминает UNIX. Каждый из них непрерывно общается со смежными маршрутизаторами и потому знает состояние своего окружения. Он знает, когда какой-то из соседей «закрыт» на техническое обслуживание или просто перегружен. Принимая решение о переправке проходящего пакета, маршрутизатор учитывает состояние своих соседей и динамически перераспределяет потоки так, чтобы пакет ушел в том направлении, которое в данный момент наиболее оптимально.

Поскольку к Интернету могут подключаться сети, работающие на основе своих, не TCP/IP, а других, протоколов, то для обмена информацией с ними используются специальные маршрутизаторы – *шлюзы* (англ. *gateways*). Опять-таки, шлюзом может быть специальный компьютер, но это может быть и специальная программа. Шлюзы выполняют преобразование данных из форматов, принятых в локальной сети, в формат, принятый в Интернете, и наоборот.

Аналогами транспортных средств (автомобилей, самолетов, поездов и т. д.), используемых в почтовых ведомствах, в Интернете являются каналы связи. Так же как и транспортные средства, каналы связи могут быть самыми различными. Они могут быть собственными или арендованными, быстрыми (цифровые спутниковые) или медленные (линии местных телефонных компаний), надежными и ненадежными и т. д.

Таким образом, в сети Интернет доставка пакетов данных от компьютера-источника до компьютера-получателя выполняется по различным каналам связи через серию маршрутизаторов на основе Межсетевого Протокола (Internet Protocol) IP, который отвечает за адресацию компьютеров в сети. Но даже хорошо разработанная система адресации не решает полностью всех проблем пересылки информации в сети.

Аналогично почтовому ведомству, при пересылке информации в Интернете возникают и такие проблемы.

- Во-первых, рассмотрим следующий пример. Как следовало бы поступить в случае, если необходимо послать кому-нибудь книгу, а почта принимает только письма? Выход один: вырвать из книги все страницы, вложить каждую в отдельный конверт и бросить все конверты в почтовый ящик. Получателю пришлось бы собирать все страницы (при условии, что ни одно письмо не пропало) и склеивать обратно в книгу. В сети Интернет, аналогично рассмотренному выше примеру, по целому ряду технических причин (в основном это аппаратные ограничения) информация так же разбивается на порции, называемые пакетами. В одном пакете обычно пересылается от одного до 1500 байт информации. Это не дает возможности одному пользователю монополизировать сеть, однако позволяет каждому рассчитывать на своевременное обслуживание. Это также означает, что в случае перегрузки сети качество ее работы несколько ухудшается для всех пользователей, но она не перестанет работать полностью, даже если некоторый ее участок будут монополизирован несколькими солидными клиентами.

- Во-вторых, может произойти ошибка. Почтовое ведомство иногда теряет письма, а сети иногда теряют пакеты или повреждают их при передаче. В Интернете, в отличие от почтового ведомства, эта проблема успешно решается.

- В-третьих, последовательность доставки пакетов может быть нарушена. Например, если по одному адресу одно за другим были посланы два письма, то нет никакой гарантии, что они пойдут по одному маршруту или придут в порядке их отправления. Такая же проблема существует и в Интернете.

Для решения всех этих задач и предназначен следующий уровень протоколов – Протокол Управления Передачей (Transmission Control Protocol) TCP, который является второй, неотъемлемой, частью семейства протоколов TCP/IP. То есть, протокол TCP отвечает за:

1. разбивку сообщения на отдельные пакеты,
2. обработку ошибок передачи пакетов, и
3. восстановления необходимой последовательности пакетов на принимающей стороне.

### 3. АДРЕСАЦИЯ КОМПЬЮТЕРОВ В СЕТИ

Каждый компьютер, подключенный к Интернет, обязательно имеет свой уникальный адрес, называемый *IP-адресом* (читается «ай-пи»). IP-адрес компьютера может быть постоянным, или каждый раз назначаться новым при соединении с сетью. Но всегда один IP-адрес соответствует только одному компьютеру.

Длина IP-адреса – 4 байта (по сетевой терминологии *октета*) или 32 бита. Для удобства эти адреса записываются в виде четырех чисел от 0 до 255, разделенных точкой.

Теоретически к сети одновременно могут быть подключено до  $2^{32} = 4\,294\,967\,296$  компьютера. На самом деле, некоторые комбинации битов зарезервированы и это число намного меньше. Так, например, адрес 127.0.0.1 используется только на локальном компьютере.

Любой IP-адрес состоит из 2-х частей:

1. *адреса сети* и
2. *адреса хоста* (*хостом* называют любой, подключенные к Интернету, компьютер).

В зависимости от того, какая часть IP-адреса выделена под адрес сети, а какая – под адрес компьютера в ней, различают сети классов *A*, *B* и *C* (существуют также служебные классы *D* и *E*). Сети класса *A* – это огромные сети, но их не много. И наоборот, сетей класса *C* очень много, но компьютеров у них мало.

Чтобы отделить адрес сети от адреса хоста, используется *маска подсети*, также представляющая собой 32-битное число. По умолчанию сетям класса *A* соответствует маска 255.0.0.0, класса *B* – 255.255.0.0, а сетям класса *C* – 255.255.255.0. То есть, в двоичном представлении маски, позиции, соответствующие адресу сети, закрыты единицами. Маска подсети может использоваться и для других целей, например, для логического разделения локальных сетей на подсети меньшего масштаба.



## 4. ДОМЕННАЯ СИСТЕМА ИМЕН

*Хост-файл*

*Создание DNS*

*Пространство доменных имен*

*Механизм разрешения*

*Регистрация доменов*

*dotCOM-бизнес*


### Хост-файл

Цифровой IP-адрес хоста удобен для его автоматической обработки компьютером, но крайне не удобен для восприятия человеком – люди предпочитают более-менее осмысленные имена. Поэтому, с самого начала существования Интернета, компьютеры помимо цифровых IP-адресов могли иметь и символьные имена. Вначале это был так называемый *хост-файл*. Хост-файл – это обыкновенный текстовый файл, который так и назывался *hosts*. Он представлял собой таблицу, каждая строка которой состояла из пары «*IP-адрес и имени хоста*». Файл *hosts* похож на телефонный справочник, где каждому номеру телефона соответствует имя владельца. Если необходимо позвонить какому-либо человеку, то по телефонной книге всегда можно найдем его номер (при условии, что он был заранее зарегистрирован). Авторство создания этого файла-таблицы принадлежит Джону Постелу (Jon Postel), который первым его и поддерживал. Для присвоения имени хосту необходимо было обратиться в специальную службу регистрации *IANA Сетевого Информационного центра (Network Information Center – NIC) в Стенфордском Исследовательском Институте (Stanford Research Institute – SRI, Menlo Park, California)* которая:

3. добавляла в файл *hosts.txt* запись с неповторяющимся именем хоста и его IP-адрес, а также

4. регулярно рассылала этот файл на все остальные компьютеры сети.

В качестве имен хостов использовались простые английские слова, каждое из которых обязательно являлось уникальным. Таким образом, поддерживалась актуальность файла *hosts* и, соответственно, имен компьютеров и их IP-адресов в сети.

 *IANA – Internet Assigned Numbers Authority принято переводить на русский как «Уполномоченная организация по распределению нумерации в сети Интернет». На самом деле, IANA – это не организация, а набор функций. То есть IANA сама по себе не имеет организационно-правовой формы. В настоящее время исполнение IANA-функций возложено на корпорацию ICANN – The Internet Corporation for Assigned Names and Numbers – Интернет-корпорация по распределению адресов и имен. К основным функциям IANA относятся:*

- координация работ по выработке технических параметров протоколов;


- административные функции по управлению корнем системы доменных имен;
- распределение блоков IP-адресов;
- прочие функции, которые могут быть включены в IANA-функции по согласованию с Министерством торговли США.


В операционной системе Microsoft Windows, например, полное имя этого файла `C:\WINDOWS\system32\drivers\etc\hosts`, а в Linux – `/etc/hosts`.

### Создание DNS

Однако, как показало время, такой подход оказался тупиковым, поскольку поддерживать актуальность хост-файла в связи с бурным ростом Интернета и, соответственно, числа компьютеров в ней, становилось все труднее и труднее. Решение этой проблемы было найдено в 1983 г., когда Пол Мокапетрис (Paul Mockapetris) из Института информатики Университета Южной Калифорнии (USC/ISI) в *RFC-882* и *RFC-883* формально описал *Доменную Систему Имен* (*Domain Name System – DNS*), которая должна была заменить изживший себя хост-файл. То есть DNS решала проблему, с которой не справилась хост-таблица, используя две концепции:

1. иерархию имен, и
2. распределение ответственности.

 *RFC – Request for Comments – документы с таким названием содержат в себе материалы по Интернет-технологиям, которые доведены до уровня стандарта или близки к этому уровню.*

 *Работа по созданию DNS была начата в 1984 г. четырьмя старшекурсниками университета в Беркли: Дугласом Терри (Douglas Terry), Марком Пойнтером (Mark Painter), Дэвидом Ригглом (David Riggle) и Сонг Ньян Чжоу (Songnian Zhou). Эстафету подхватил Ральф Кэмпбелл (Ralph Campbell) из Computer Systems Research Group, который начал «склеивать» доменную систему имен в BSD-UNIX. В 1985 г. Кевин Данлэн (Kevin Dunlap), инженер DEC, временно работавший в Беркли, принял этот проект в свои руки и создал систему BIND (Berkeley Internet Name Domain – систему доменных имен Интернета реализации Беркли). Майк Кареле (Mike Karels) и Пол Викси (Paul Vixie) сопровождали эту систему в течение ряда лет. Пол продолжает вести ее и сейчас, пользуясь помощью участников телеконференции *isc.org* и членов списка рассылки *bind-workers*.*

DNS – это, прежде всего, один из ключевых элементов инфраструктуры Интернета, который обеспечивает трансляцию алфавитно-цифровых имен компьютеров в их цифровые IP-адреса, необходимые для физического доступа к ним. Фактически DNS выполняет функции «человеко-машинного

интерфейса» между пользователями Интернета с одной стороны и системами адресации и маршрутизации с другой стороны.

Доменная система имен, как один из множества сервисов (служб) Интернета, выполняет две основных функции:

1. организацию *пространства доменных имен*, и
2. обеспечение *механизма разрешения*, т. е. сопоставление символического имени компьютера его цифровому IP-адресу.

### Пространство доменных имен

Пространство доменных имен (*Domain Name Space*) сменило плоскую систему именования компьютеров в сети, реализованную при помощи хост-файла, на иерархическую структуру, допускающую наличие в имени произвольного количества составных частей. Фрагмент такой структуры представлен на Рис. 5.2.

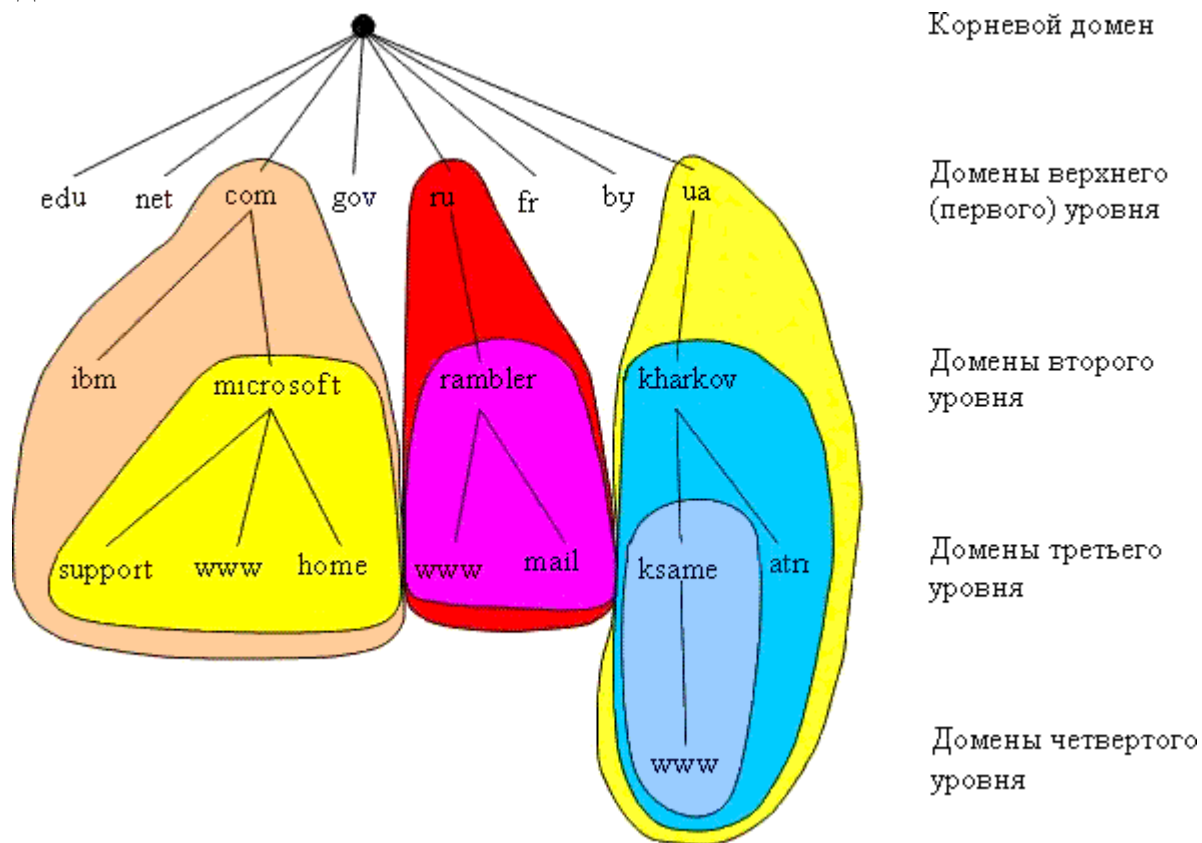


Рис. 5.2. Фрагмент пространства доменных имен

Пространство доменных имен аналогично иерархии имен файлов, принятой в большинстве файловых систем. Оно *начинается с корня, который имени не имеет* и называется *корневым доменом* или *доменом нулевого уровня*. Затем следуют домены *первого уровня*, *второго уровня* и т. д. уровней.

Последний уровень пространства доменных имен (лист) соответствует конечному узлу сети (хосту) – реальному компьютеру с соответствующим IP-адресом. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д.,

запись доменного имени начинается с самой младшей составляющей, а заканчивается – самой старшей. Составные части доменного имени друг о друга отделяются точкой. Например, в имени `www.ksame.kharkov.ua` составляющая `www` является именем одного из компьютеров в домене `kname.edu.ua`.

Имена компьютеров бывают двух видов:

1. собственные и
2. функциональные.

Собственное имя компьютера назначается его владельцем или администратором домена произвольно. Функциональные имена вытекают из функций, которые он выполняет. Например:

`www` – сервер Всемирной Паутины (WWW),  
`ftp` – FTP-сервер,  
`mail` – почтовый сервер и т. д.

Домены первого уровня (*Top Level Domains – TLDs*) делятся на две категории:

1. домены общего назначения (*Generic TLDs – gTLDs*) и
2. национальные домены (*Country Code TLDs – ccTLDs*).

Домены общего назначения (организационные, родовые – в зависимости от рода деятельности) – это, как правило, трехбуквенные имена, предназначены для использования всем Интернет-сообществом. Наиболее широко используемыми (наиболее «старыми») доменами общего назначения являются следующие:

Название	Сокращение от ...	Род деятельности (использование)
<code>com</code>	<b>commercial</b>	Коммерческие
<code>edu</code>	<b>educational</b>	Образовательные
<code>gov</code>	<b>government</b>	Правительственные
<code>mil</code>	<b>military</b>	Военные
<code>net</code>	<b>network</b>	Обеспечивающие работу сети
<code>org</code>	<b>organization</b>	Некоммерческие

Изначально `gTLDs` предназначались для объединения доменов нижних уровней, принадлежащих организациям и учреждениям США. Однако ничто не мешает какой-нибудь компании, например, российской или украинской, зарегистрировать свой домен второго уровня в домене «`com`». Исключением являются только домены «`mil`» и «`gov`», которые используются только учреждениями и организациями США.

Национальные (региональные, географические) домены – это двухбуквенные имена, которые обозначают коды стран в соответствии со стандартом ISO 3166-1. Например, «`ua`» – Украина, «`ru`» – Россия, «`by`» – Беларусь, «`fr`» – Франция и т. д. Для США наименование страны по традиции опускается – поскольку Интернет зародилась и развивалась в США – там самыми крупными объединениями являются домены общего назначения (`gTLDs`). Однако, для обеспечения единообразия, США имеют свой

национальный домен – «us». Для бывшего СССР также был выделен домен – «su», который поддерживается, но новые домены более низких уровней в нем больше не регистрируются.

Каждый домен верхнего уровня включает в себя домены (*поддомены*) второго уровня, имена которых выбираются относительно произвольно, например, по имени компании, за которой зарегистрировано это имя, или по названию региона. Порядок создания доменов второго уровня определяется администратором соответствующего *родительского* домена первого уровня.

В доменах (зонах) государств, опять же, могут быть и организационные и географические зоны. Организационные домены в большинстве своем повторяют структуру организационных доменов верхнего уровня. Географические зоны выделяются по городам, областям и другим территориальным образованиям. Непосредственно и в тех и в других размещаются домены третьего уровня, обычно относящиеся к организациям, подразделениям внутри компаний или домены персональных пользователей.


Общие правила построения доменных имен следующие:

- имя может состоять только из букв латинского алфавита, цифр и символа «-» (дефис);
- длина каждой составляющей доменного имени не может превышать 63 символов;
- общая длина доменного имени не может превышать 255 символов;
- отдельные составляющие доменного имени друг от друга отделяются символом точка «.»;
- доменные имена являются нечувствительными к регистру символов, входящих в его состав, например, последовательности символов «Com», «COM», «cOm», «com» и т. п. обозначают одно и то же имя; реализации DNS должны игнорировать регистр символов при сравнениях, но обязаны распространять его, если он указан.

Разделение доменного имени на части позволяет разделить административную ответственность за назначение уникальных имен в пределах своей компетенции. Так, например, администратор домена «ua» несет ответственность за то, чтобы все имена в его зоне ответственности были неповторяющимися. Это позволяет решить проблему образования уникальных имен без взаимных консультаций – если в каждом домене обеспечивается уникальность имен следующего уровня иерархии, то и вся система доменных имен будет состоять из уникальных имен. Таким образом, каждый компьютер (узел, хост) в сети Интернет однозначно определяется своим *полным доменным именем* (*Fully Qualified Domain Name – FQDN*), которое включает имя хоста и имена всех доменов по направлению от узла (листа) к корню дерева. Например,

(Рис. 5.2.) компьютер имеет имя «mail» (почтовый сервер). Если собрать имена всех доменов по структуре от листа до корня дерева, то получится полное доменное имя – «mail.rambler.ru». По аналогии с файловой системой в пространстве доменных имен различают также *краткие* и *относительные имена*. Краткое имя – это имя конечного узла сети (хоста, листа), например, (Рис. 5.2) «atn» в домене «kharkov.ua». Относительное имя – это составное имя, начинающееся с некоторого уровня иерархии, но не с самого верха. Например, (Рис. 5.2) «www.ksame» в домене «kharkov.ua».

Таким образом, доменное пространство имен – это метод назначения символических имен хостам путем передачи ответственности сетевым группам за их подмножество имен. В результате адрес того или иного ресурса Интернет, записанный в стандарте DNS, дробится на несколько составляющих, отделенных друг от друга точкой. Каждый из этих элементов, кроме самого левого, носит название *домен* (англ. *domain* – область, сфера, зона и т. д.). То есть, домен представляет собой некое логическое подмножество сетевых ресурсов, имеющее собственное имя и управляемое из единого центра.

 Следует иметь в виду, что доменные имена в реальной жизни достаточно причудливо отображаются на IP-адреса реальных компьютеров, которые физически подключены к сети. Так, например, компьютер, физически установленный и подключенный к Интернету в далекой Америке, может совершенно спокойно иметь имя из украинского корпоративного домена, например, *www.salo.ua*, и наоборот, компьютер украинского сегмента может иметь имя из домена «com». Последнее, к слову сказать, встречается гораздо чаще.

Более того, один и тот же компьютер может иметь несколько доменных имен; и наоборот – возможен вариант, когда за одним доменным именем может быть закреплено несколько IP-адресов, которые реально назначены различным компьютерам. Таким образом, соответствие между доменными именами и IP-адресами в рамках системы доменных имен не является взаимно однозначным, а строится по схеме **«многие-ко-многим»**.

Доменное пространство имен, возможно, и выглядит несколько сложновато, но это одна из составляющих, которая делает общение с сетью более простым и удобным. Несомненное преимущество доменного пространства имен состоит в том, что оно разбивает Интернет на набор вполне обозримых и управляемых частей. Хотя сеть включает миллионы компьютеров, все они поименованы, и именованное это организовано в удобной и рациональной форме.



## Механизм разрешения


Механизм разрешения, т. е. сопоставление символического имени компьютера его цифровому IP-адресу в сети, построен по схеме «клиент-сервер» и состоит из двух частей:

1. DNS-клиента (*name-resolver*) – программы, выполняющейся на компьютере клиента, которой требуется найти по доменному имени IP-адрес компьютера, и

2. DNS-сервера (*name-server*) – программы, осуществляющей по запросу DNS-клиента поиск IP-адреса на основе предложенного ей доменного имени.

Параметры программного обеспечения DNS-клиента настраиваются во время подключения компьютера к сети. Так, при подключении компьютера к локальной сети в параметрах сетевого подключения стека протоколов TCP/IP необходимо указать IP-адрес первичного (предпочитаемого) локального сервера имен, а также, возможно, вторичного (альтернативного) DNS-сервера. При непосредственном же подключении компьютера к Интернету, адреса первичного и вторичного DNS-серверов могут быть настроены как вручную, так и автоматически. То есть, в последнем случае локальным сервером имен будет DNS-сервер Интернет-провайдера.

DNS сервера – это компьютеры, на которых установлено специальное программное обеспечение. Каждый DNS-сервер обслуживает свою *зону ответственности* в иерархическом пространстве доменных имен.

 *Часть иерархического пространства имен DNS, обслуживаемая сервером имен и представленная в его локальной базе данных, называется зоной ответственности (Zone Of Authority).*

Важным преимуществом системы доменных имен является то, что каждый конкретный DNS-сервер имен не должен содержать в своей локальной базе данных описание всей иерархии пространства доменных имен. В его базе данных представлено только пространство имен, принадлежащих его домену. Например, (Рис. 5.2) если в домене «kharkov.ua» имеется только два узла – «ksame» и «atn», то в базе данных сервера имен этого домена может присутствовать всего две записи для указанных имен. Кроме того, если домен содержит в себе домены нижних уровней, то каждый такой поддомен (*субдомен*) может иметь свой собственный сервер имен, освобождая тем самым от необходимости обслуживать свое подпространство имен DNS-сервер родительского домена. Например, (Рис. 5.2) поддомен «ksame» в домене «kharkov.ua». Таким образом, продолжая пример (Рис. 5.2), сервер имен домена «ua» не имеет в своей базе данных информации об узлах «ksame» и «atn» домена «kharkov.ua». Такая передача полномочий по управлению именами части зоны ответственности называется *делегированием*.

Таким образом, DNS-сервера представляют собой распределенную иерархическую базу данных, которая содержит информацию обо всех компьютерах (хостах), подключенных к сети Интернет. Для нормального функционирования системы доменных имен каждому DNS-серверу не обязательно знать адреса всех остальных DNS-серверов Интернет. Для этого ему достаточно располагать IP-адресами:

1. корневых DNS-серверов, а также
2. всех DNS-серверов делегированных зон (поддоменов, субдоменов).

Для распределения нагрузки, а также повышения отказоустойчивости всей системы доменных имен в каждой зоне ответственности одновременно и параллельно функционирует несколько DNS-серверов. То есть DNS-сервера не существуют в единственном экземпляре. У каждого из них (еще называют первичным, основным, предпочитаемым DNS-сервером) есть один или несколько «братьев-близнецов» (вторичных, альтернативных DNS-серверов). Вся информация, записанная на первичном DNS-сервере, периодически копируется на все остальные. Это дает возможность нормально работать всей системе DNS в случае выхода из строя любого из серверов. Так, за поддержку самого верхнего, корневого, домена на сегодняшний день отвечают тринадцать *корневых DNS-серверов (Root Servers)*. Десять из них расположены в США, два – в Европе и один – в Японии. На них, прежде всего, содержится информация о серверах, которые поддерживают домены первого уровня.

Алгоритм работы службы DNS достаточно прост. Когда программа-клиенту требуется по доменному имени выяснить IP-адрес, она связывается с локальным DNS-сервером имен, адрес которого указан в настройках сетевого протокола TCP/IP. Локальный сервер имен, получив такой запрос, рассматривает его, чтобы выяснить, в каком домене находится указанное имя. Если указанный домен входит в его зону ответственности, то сервер преобразует имя в IP-адрес на основе собственной базы данных и возвращает результат клиенту. Если же запрашиваемое доменное имя не входит в его базу данных, то он переадресует запрос вышестоящему DNS-серверу. В любом случае по указанному доменному имени программа-клиент всегда получит необходимый ей IP-адрес.

Таким образом, доменная система имен позволила создать масштабируемый распределенный механизм для отображения иерархических имен компьютеров в их цифровые IP-адреса.



## Регистрация доменов

Зарегистрировать доменное имя можно двумя способами:

1. самостоятельно, изучив инструкции на соответствующих серверах, или
2. обратиться к поставщику услуг Интернет, который возьмет на себя все хлопоты по регистрации доменного имени.

В последнем случае главное необходимо проследить, чтобы домен был зарегистрирован именно на Вас или Вашу компанию, а не на Интернет-провайдера. Обработка заявки на регистрацию домена занимает около двух недель.

Процессом распределения IP-адресов и доменных имен *первого уровня* управляет неправительственная, некоммерческая организация ICANN (The Internet Corporation for Assigned Names and Numbers – Интернет-корпорация по распределению адресов и имен) имеющая адрес [www.icann.org](http://www.icann.org). Полномочия по распределению доменных имен в зонах *второго уровня* ICANN передает другим уполномоченным организациям – *регистраторами*. Регистраторы, в свою очередь, могут предоставлять другим организациям возможность регистрировать домены через свои партнерские программы. Регистрацию доменов второго уровня на территории США осуществляет InterNIC (Internet Network Information Center), которая находится по адресу [www.internic.net](http://www.internic.net). В Европе ее функцию взяла на себя организация RIPE (фр. Réseaux IP Européens), имеющая адрес [www.ripe.net](http://www.ripe.net). В России регистрацией доменов в зоне «ru» занимается RIPN с адресом [www.ripn.net](http://www.ripn.net).

Найти другого регистратора в каком-либо конкретном региональном или международном домене тоже не сложно. В региональных доменах, обычно, есть сайты с адресами вида «[www.nic.domain](http://www.nic.domain)». То есть, для украинского домена это будет «[www.nic.ua](http://www.nic.ua)», для российского – «[www.nic.ru](http://www.nic.ru)», а для белорусского – «[www.nic.by](http://www.nic.by)». Если же подходящего сайта не оказалось, можно просто выполнить поиск вида «регистрация доменов ua» и сразу получить список многих регистраторов украинского домена (запрос желательно писать на соответствующем языке).

Регистрация домена, чаще всего, платная, т. к. и корневые сервера и сервера, обслуживающие региональные домены, требуют технической поддержки. То есть домен покупается. Тем самым приобретается право владения доменом на какой-то определенный срок (минимум 1 год), подлежащий продлению (также платному). Владение доменом подразумевает, что его владелец имеет право полностью распоряжаться тем, какие сервера будут обрабатывать конечные запросы в этом домене, т. е. будут *авторитетными* для данного домена.

Организации или физическому лицу, желающим зарегистрировать свой домен, следует обращаться к администратору какого-либо уже существующего

домена. Однако в любом случае предварительно необходимо проверить, не зарегистрировано ли уже такое имя. Это можно сделать следующим образом. В большинстве зон ответственности имеется свой сервер «whois», через который можно узнать подробную информацию о человеке или организации, на которую зарегистрирован тот или иной домен. Например, в зоне «ua» имеется сервер whois.com.ua. Он также отвечает за домены второго уровня, такие как «net.ua», «com.ua» и т. д. Для зоны «ru» информацию о доменном имени можно получить также по адресу [www.ripn.net/nic/whois/](http://www.ripn.net/nic/whois/), а для доменов «com», «org», «net», «edu» и т. д. – по адресам [www.register.com](http://www.register.com) или [www.internic.net/whois.html](http://www.internic.net/whois.html). Если же выбранное имя уже зарегистрировано, то остается попытаться придумать другое. Также можно попробовать выйти на организацию или частное лицо, владеющее данным доменом, и попытаться его перекупить.

### **dotCOM-бизнес**

С точки зрения технологии сетей TCP/IP, на которых строится межсетевое взаимодействие в Интернете, DNS является вспомогательной прикладной службой. Однако значение DNS для нужд навигации в Интернете трудно переоценить. Оно выходит далеко за пределы простого преобразования «имя–адрес».

Целая отрасль экономики *dotCOM* получила название от доменного имени «.com». Таким образом, люди, в руках которых находится управление системой DNS или отдельными ее компонентами, обладают «магической» властью. По их неосторожности или злему умыслу из Интернета могут «исчезнуть» не только отдельные компании или персоны, но и целые страны, как это произошло, например, 9 апреля 2004 года с доменным именем Ливии.

Доменное имя в новых условиях стало выполнять функции бренда, торговой марки, наименования товара и т.д. Это, в свою очередь, привело к борьбе за право управления соответствующими доменами; возникла необходимость создания законодательной базы для решения таких споров. Взрывной рост количества коммерческих информационных ресурсов в Интернете в период с 1994 по 1998 год позволил перевести поддержку инфраструктуры Интернета на самоокупаемость. Управление серверами, реестрами и регистрация доменных имен стали самостоятельными видами бизнеса.

## **5. ЕДИНООБРАЗНЫЙ ЛОКАТОР РЕСУРСА**

Единообразный локатор (определитель местонахождения) ресурса (*URL – Uniform Resource Locator*), который ранее назывался *Universal Resource Locator* – универсальный локатор ресурса – это стандартизированный способ записи адреса ресурса в сети Интернет. URL был разработан Тимом Бернерсом-Ли в 1990 году, как одна из составляющих Всемирной паутины (WWW). Изначально

он предназначался для максимально естественного указания мест расположения во Всемирной паутине таких ресурсов, как Web-страницы.

URL состоит из двух частей:

1. названия схемы доступа к ресурсу (обычно – название протокола) и
2. специфической для данной схемы информации, например, название и местоположения ресурса, а также метод доступа.

В целом же синтаксис URL представляет собой расширение принятого на локальных компьютерах понятия *полного имени файла* применительно к множеству ресурсов Интернета и выглядит следующим образом:

[<протокол>://][<логин>:<пароль>@][<хост>][:<порт>][/<путь>][<имя файла>][<дополнительная информация>]

Компоненты в этой записи являются необязательным (все они помещены в квадратные скобки []) и выполняют такие функции:

**протокол** – сетевой протокол (название схемы доступа) для обращения к ресурсу является первой частью URL и определяет его вторую, специфическую, часть. Наиболее распространенными протоколами в сети Интернет являются следующие:

*http* – протокол передачи гипертекста,

*https* – специальная реализация протокола HTTP, использующая шифрование (как правило, SSL или TLS),

*mailto* – адрес электронной почты,

*ftp* – протокол передачи файлов,

*telnet* – интерактивная сессия Telnet,

*file* – имя файла на локальном компьютере,

*data* – непосредственные данные.

От остальной части URL протокол отделяется двоеточием (:) и двумя наклонными линиями – двойным слешем (//).

**логин** – имя пользователя, используемое для доступа к ресурсу.

**пароль** – пароль, соответствующий указанному имени пользователя.


Логин от пароля отделяется символом «двоеточие» (:). Они используются не со всеми протоколами. Например, с протоколом telnet они применяются для «входа» практически в любой компьютер Интернета, а с протоколом HTML – никогда, поскольку могут быть использованы злоумышленниками. Если же логин и пароль присутствуют, то от оставшейся части URL они отделяются символом «собачка» (@).

**хост** – доменное имя хоста или его IP-адрес.

**порт** – это некоторое условное число в диапазоне от 1 до  $2^{16} - 1 = 65535$ , позволяющие различным программам, выполняющимся на одном компьютере, получать данные независимо друг от друга. Каждая программа обрабатывает

данные, поступающие на определенный порт (иногда говорят, что программа «слушает» этот номер порта). Как правило, в URL номер порта явно не указывается – используется значение по умолчанию, например, значение 80 для протокола HTTP, 23 – для telnet, 21 – для команд, а 20 – для данных в протоколе FTP и т. д.

**путь** – задает папку (каталог) на сервере, в которой содержится необходимый ресурс (как правило, файл).

 Поскольку большинство серверов Интернет работают под управлением операционной системы UNIX (или ей подобной), то для разделения отдельных папок в «пути» используется прямой слеш (/), а не обратный (\), как принято в Microsoft Windows.

**имя файла** – имя файла, в котором находится данный ресурс.

**дополнительная информация** – в качестве дополнительной информации могут выступать, например:

а) идентификатор фрагмента файла – для непосредственного доступа к конкретному фрагменту HTML-документа, обозначенного меткой («якорем»). В этом случае метке должен предшествовать символ «решетка» (#).

б) параметр – для передачи дополнительной информации в программу, которая вызывается в качестве ресурса на сервере в соответствии с общим интерфейсом шлюзов CGI. В этом случае параметру должен предшествовать символ «вопросительный знак» (?).

При записи URL необходимо учитывать, следующее:

- могут использоваться только буквы латинского алфавита и некоторые символы, такие, например, как «тире» (-), «подчеркивание» (\_), «тильда» (~) и т. д.

- если же необходимо использовать символы кириллицы, то каждый из них необходимо представить в Юникоде (UTF-8) в виде последовательности из двух байтов в шестнадцатеричном представлении, перед каждым из которых ставится знак «процента» (%); например, большая русская буква «М» будет записана как «%D0%9C»;

- нельзя использовать символ «пробел»;

- прописные и строчные символы, входящие в его состав, различаются (за исключением компонента **хост**, если он задан в виде доменного имени).

Ниже приведены наиболее типичные примеры записи URL реальных Web-страниц:

1. <http://www.ksame.kharkov.ua/index.php> – наиболее распространенный способ записи URL, в котором указаны протокол (http), доменное имя (www.ksame.kharkov.ua) и имя файла (index.php) – начальная страница сайта Харьковского национального университета городского хозяйства им. А. М. Бекетова.

2. [www.ksame.kharkov.ua](http://www.ksame.kharkov.ua) – аналог предыдущего URL, в котором опущены протокол и имя файла. Поскольку, имя компьютера (www) в доменном имени явно указывает на его принадлежность к Всемирной Паутине, то протокол http будет подставлен автоматически. Если же не указано имя файла, то по умолчанию серверы WWW настроены на выдачу страниц типа *index.html*, *index.htm* или *index.php* – как в данном примере.

3. <http://www.ksame.kharkov.ua/portal/index.php?mnu=33> – URL с указанием дополнительной информации (?mnu=33) – страница с расписанием занятий студентов на Web-портале Харьковского национального университета городского хозяйства им. А. М. Бекетова.


4. <http://ru.wikipedia.org:80/wiki/URL> – URL с явным указанием номера порта (80) – описание Единообразного локатора ресурса на сайте свободной энциклопедии Википедия.

5. <ftp://ftp.microsoft.com/> – URL с ftp протоколом – «подвал» корпорации Microsoft – обширное хранилище полезных программ, предназначенных для скачивания.

6. <mailto://MyMail@ksame.kharkov.ua> – URL адреса (несуществующего) электронной почты – протокол mailto.

7. <telnet://www.ksame.kharkov.ua> – подключение к Web-сайту Харьковского национального университета городского хозяйства им. А. М. Бекетова по протоколу telnet (необходимы так же логин и пароль).

8. <file://D:\course1\БУА-1\Иванов\HTML\index.html> – URL для просмотра файла *index.html*, расположенного на локальном компьютере.

 *Использование символов кириллицы (русских) и обратного следа (\) вместо прямого (/) в данном примере возможно, поскольку компьютер работает под управлением локализованной версии Microsoft Windows.*

В последнее время URL позиционируется как часть более общей системы идентификации ресурсов – *URI – Uniform Resource Identifier* – универсальный индикатор ресурса. И сам термин URL постепенно уступает место более широкому термину URI.

Полное описание стандарта URL, одобренного комитетом W3C, содержится в RFC 1738.

## ЛЕКЦИЯ №6

### ОСНОВЫ ВЕБ-ДИЗАЙНА

#### План

1. Назначение языка HTML
2. Понятие Web-страницы и Web-сайта
3. История создания языка HTML
4. Структура HTML-документа
5. Инструментарий для создания HTML-документов

#### 1. НАЗНАЧЕНИЕ ЯЗЫКА HTML

Наиболее востребованным на сегодняшний день сервисом Интернета является всемирная паутина – World Wide Web (WWW). Она «соткана» из множества отдельных Web-страниц, каждая из которых состоит из:

1. содержания,
2. представления и
3. поведения.

Содержание Web-страницы – это текст, графика, видео и аудио – одним словом, это информация, которая помещается на страницу. Для описания структуры этой информации служит язык *HTML (Hyper Text Markup Language)* – язык гипертекстовой разметки). Представление описывает формат вывода этой информации. Именно представление делает Web-страницы такими привлекательными. Представление Web-страниц реализуется при помощи каскадных таблиц стилей (*Cascading Style Sheets – CSS*). Поведение – это динамическое изменение Web-страницы в зависимости от действий пользователя или каких-либо иных событий. Оно задается с помощью так называемых *сценариев (скриптов)* – программ, которые могут быть записаны прямо в коде Web-страниц. Такие программы создаются на одном из скриптовых языков. Чаще всего – на *JavaScript*.

Основой любой Web-страницы является ее содержание – без содержания страница существовать не может. Оно описывается (размечается) при помощи языка HTML. То есть, с помощью языка HTML описываются различные фрагменты документа (текст, рисунки, гиперссылки и т. д.), а так же их взаимное расположение.

Для просмотра документов, размеченных по правилам языка HTML, предназначены специальные программы – *браузеры (browsers)*. Они читают файл HTML-документа, форматируют его в виде Web-страницы и отображают на мониторе пользователя. Существует большое разнообразие программ-браузеров от различных производителей. Однако, на сегодняшний день, наиболее распространенными являются Internet Explorer корпорации Microsoft, а также Google Chrome разработанный в корпорации Google Inc, Mozilla Firefox от Mozilla Corporation и Opera от Opera Software.

## **2. ПОНЯТИЕ WEB-СТРАНИЦЫ И WEB-САЙТА**

Web-страницей обычно называют отдельный HTML-документ, размеченный при помощи языка HTML и предназначенный для публикации в сети Интернет. Набор таких Web-страниц, связанных общей тематикой, называют Web-сайтом.

Одна из этих страниц является основной (головной, стартовой, индексной и т. д.) и автоматически выдается на просмотр пользователю, указавшему в браузере только имя сайта. Все остальные страницы сайта, как правило, выдаются пользователю из основной при помощи гиперссылок.

Каждый отдельно взятый Web-сайт, как правило, кроме общей тематики, отличается еще и общим дизайном, т. е. общий вид и компоновка страниц. Это позволяет посетителям сайта легко на нем ориентироваться, так как страницы выполнены в едином стиле, на всех страницах одинаковая или похожая навигация, а меняется только их содержание.

## **3. ИСТОРИЯ СОЗДАНИЯ ЯЗЫКА HTML**

Начало развития языка HTML было положено выпускником Оксфордского университета, бакалавром в области физики, Тимом Бернерс-Ли (Tim Berners Lee). В 1989 году он выдвинул предложение о *Системе гипертекстовых документов* для создания внутренней информационной службы Европейского центра ядерных исследований (CERN). В 1990 году он назвал ее World Wide Web (на русский язык это можно перевести как Всемирная Паутина). Одной из составляющих этой системы был язык гипертекстовой разметки. Основные принципы языка HTML были заложены Тимом Бернерс-Ли в том же 1990 году, когда он работал над программой первого текстового Web-браузера. За основу им был взят язык для логического оформления текста SGML (Standard Generalized Markup Language), стандарт которого был принят Международной организацией по стандартизации (ISO) в 1986 году.

А первым (и долгое время единственным) графическим браузером в те далекие времена была программа Mosaic, разработанная, как и сама World Wide Web, в научном учреждении – Национальном центре суперкомпьютерных приложений США (National Center for Supercomputer Applications – NCSA) группой программистов возглавляемой Марком Андресеном в сентябре 1993 года. Быстроте, с которой новинка покорила виртуальное пространство, может позавидовать иной вирус – всего за год около двух миллионов пользователей установили Mosaic на свои компьютеры. Именно благодаря популярности браузера Mosaic язык HTML получил такое распространение.

В том же 1993 году появилась первая версия языка – HTML 1.0. Но поскольку она не была стандартизована, то разные производители браузеров

стали предлагать свои варианты инструкций HTML, число которых все возрастало и возрастало. А для существования сети необходимо, чтобы авторы Web-страниц и производители браузеров пользовались одними и теми же соглашениями. Это и послужило причиной для начала совместной работы по созданию спецификации языка HTML. Такую работу взяла на себя организация, называемая World Wide Web Consortium (сокращенно W3C). В ее задачу входило составление спецификации, отражающей текущий уровень развития возможностей языка с учетом разнообразных предложений компаний-разработчиков браузеров. Так, в ноябре 1995 года появилась спецификация HTML 2.0, призванная формализовать сложившуюся к концу 1994 года практику использования HTML. К тому времени новую версию языка HTML полностью поддерживало большинство браузеров. Последней версией языка, которая стала стандартом в 1999 году, является HTML 4.01. Следующая версия языка – HTML 5 – находится на стадии разработки. Однако большинство современных браузеров уже сейчас поддерживают его в том или ином виде.

#### 4. СТРУКТУРА HTML-ДОКУМЕНТА

HTML-документ представляет собой обыкновенный *текстовый файл* в формате ANSI ASCII, который состоит из 2-х типов объектов:

1. объекты, которые должны отображаться на Web-странице (текст, графика, мультимедиа и т. д.) и
2. объекты, которые управляют расположением и отображением на Web-странице первых.

Объекты, которые управляют расположением и отображением на Web-страницах текста, графики и т. д., а сами на экране не отображаются, в языке HTML принято называть *тегами* (от английского слова *tag* – ярлык, признак). Теги вместе с объектами (фрагментами документа), отображением которых он управляет, называются *элементами* Web-страницы. Формальный синтаксис элемента представлен на Рис. 6.1.

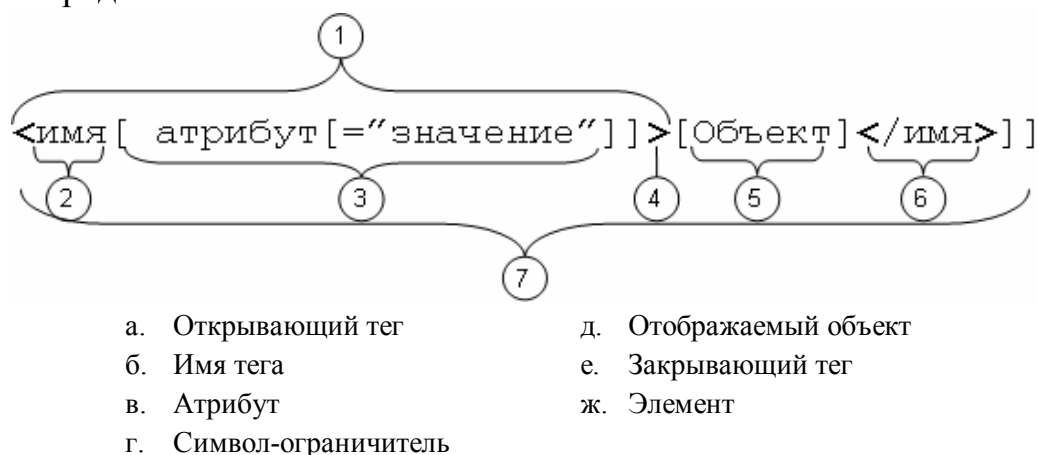


Рис. 6.1. Элемент HTML-документа



 Составляющие в квадратных скобках «[]» являются не обязательными.

Для отделения тегов от других объектов HTML-документа они заключаются в символы-ограничители «<» и «>» (поз. 4. Рис. 6.1), между которыми записываются *имя* тега (поз. 2. Рис. 6.1) и, возможно, его *атрибуты* (поз. 3. Рис. 6.1). Имя тега указывает на характер выполняемых им действий по разметке и форматированию, а атрибуты (параметры) уточняют способ выполнения этих действий. Многие атрибуты требуют указания их значений, которые следуют за знаком равенства, стоящим после имени атрибута. Если значение атрибута – одно слово или число, то его можно просто указать после знака равенства, не выделяя дополнительно. Все остальные значения необходимо заключать в одинарные «'» или двойные кавычки «"», особенно если они содержат несколько разделенных пробелами слов. Длина значения атрибута ограничена 1024 символами. Некоторые атрибуты значений не имеют или могут быть записаны без них. В последнем случае они принимают некоторые значения по умолчанию. Атрибуты являются не обязательными составляющими тега и если они имеются, то отделяются от имени, и друг от друга, по крайней мере, одним пробелом. Порядок следования атрибутов тега произволен.

Большинство тегов языка HTML являются контейнерами, т. е. отображаемый объект (поз. 5. Рис. 6.1) помещается между открывающим (поз. 1. Рис. 6.1) и, соответствующим ему, закрывающим (поз. 6. Рис. 6.1) тегами. Закрывающий тег имеет такие особенности:

5. имя закрывающего тега идентично имени открывающего;

6. перед именем закрывающего тега всегда ставится наклонная черта (прямой слеш) «/»;

7. закрывающий тег не имеет атрибутов (параметров).

Однако не все теги являются контейнерами – некоторые состоят только из открывающего, например, <br> – перевод строки. Такие элементы называются *пустыми*.

В языке HTML регистр символов в именах тегов и атрибутов не учитывается. Поэтому их можно записывать как строчными, так и прописными буквами, или их сочетанием. Например, правильно будут обработаны такие теги, как <br>, <BR> и <Br>. Однако, при записи значений атрибутов регистр символов следует учитывать.

Среди множества тегов языка HTML есть несколько *обязательных*, т. е. тех, которые всегда должны присутствовать в любом *действительном* HTML-документе. Их немного – всего четыре: <html>, <head>, <title> и <body> – вместе они создают структуру HTML-документа (Рис. 6.2).

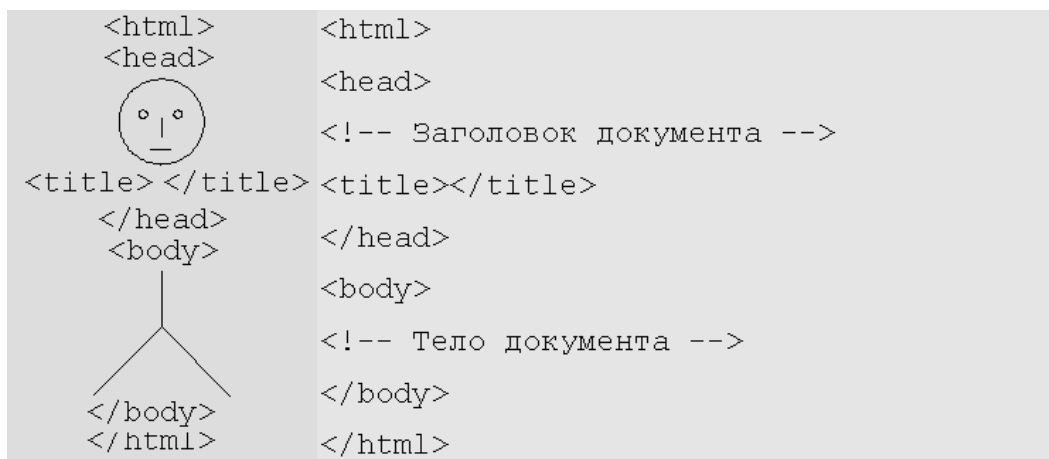


Рис. 6.2. Структура HTML-документа

Любой HTML-документ – это большой `html` элемент-контейнер. Его первым, открывающим, тегом должен быть `<html>` и, соответственно, последним, закрывающим – `</html>`. Этот контейнер, в свою очередь, содержит два элемента:

1. *head* (заголовок), который оформляется при помощи тегов `<head>` и `</head>`,
2. *body* (тело), которое оформляется при помощи тегов `<body>` и `</body>`.

В заголовке HTML-документа содержится в основном служебная информация о нем, тогда как тело документа содержит непосредственно сам документ – текст, рисунки и прочие элементы, которые отображаются на экране. Элементы, находящиеся внутри заголовка, браузером не отображаются. Исключение составляет элемент `title` – название документа, которое отображается в строке заголовка браузера.

Поскольку большинство тегов языка HTML являются контейнерами, то помимо отображаемых объектов они могут содержать и другие теги, т. е. теги могут быть вложенными друг в друга. При этом необходимо соблюдать определенный порядок открывающих и закрывающих тегов: *тег, который был открыт первым, закрывается последним, а последний – первым*. То есть, такая последовательность тегов:

`<тэГ1><тэГ2> ... </тэГ2></тэГ1>` – правильная,

а такая:

`<тэГ1><тэГ2> ... </тэГ1></тэГ2>` – нет.

HTML-документ помимо тегов может содержать так же *комментарии*. Они ни как не влияют на отображение документа, а служат для пояснения отдельных его фрагментов. Например, если над документом работает несколько человек, или для того, чтобы через некоторое время было легче вспомнить и понять свой собственный текст. Комментарии так же иногда используются, когда необходимо временно исключить некоторый фрагмент кода из обработки, не удаляя его совсем. Для этого необходимо заключить такой фрагмент в комментарии, после чего браузером он будет игнорироваться.

Хотя текст комментариев и не отображается браузером, однако передается вместе с документом и вполне может быть просмотрен. Так происходит потому, что большинство браузеров предоставляют возможность просмотра исходного кода документа. Поэтому не следует включать в комментарии информацию, не предназначенную для чужих глаз.

Комментарии в языке HTML имеют следующий синтаксис:

<!-- Текст комментария -->

То есть, синтаксически это пустой элемент, имя которого «!--», а семантически (по смыслу) он, конечно же, отличается от любого тега.

## 5. ИНСТРУМЕНТАРИЙ ДЛЯ СОЗДАНИЯ HTML-ДОКУМЕНТОВ

Для создания любого документа нужна соответствующая программа, и Web-страницы не являются исключением. Существует различные способы создания таких страниц. Простейший из них состоит в том, что некоторые программы (такие, например, как Microsoft Word, Microsoft Excel, OpenOffice.org Writer, OpenOffice.org Calc, LibreOffice Writer, LibreOffice Calc и др.) «умеют» сохранять свои документы в формате .html. Для этого необходимо:

1. подготовить нужный документ,
2. выполнить команду **Файл ⇨ Сохранить как** и
3. в раскрывшемся окне диалога указать, что файл необходимо сохранить в формате HTML-документа.

Основные достоинства этого метода состоят в том, что Web-страницы создаются довольно быстро и просто, а так же совершенно не требуется знание языка HTML. Но на этом достоинства этого метода и заканчиваются – начинаются его недостатки. Наиболее существенные из них состоят в том, что страницы получаются «массового производства», их размеры в десятки раз превышают аналогичные, но созданные «вручную», изменять такие страницы в дальнейшем весьма проблематично. Соответственно, этот способ используется, в основном, в экстренных случаях – т. е. когда катастрофически не хватает времени, а срочно необходимо один раз быстро создать несколько страниц, которые в дальнейшем не подлежат изменению.

Более совершенный способ создания Web-страниц состоит в использовании специальных программ – HTML-редакторов, которые бывают двух видов:

1. визуальные и
2. текстовые.

При использовании визуальных редакторов знания языка HTML является не обязательным. Они позволяют работать с Web-страницей «как она есть». То есть, пользователь редактирует и форматирует текст, вставляет рисунки, таблицы, как в обычном текстовом редакторе, а уж сама программа генерирует

соответствующий HTML-код. Именно поэтому визуальные редакторы еще называют WYSIWYG-редакторами. Аббревиатура WYSIWYG расшифровывается как What You See Is What You Get – «что видишь, то и получишь». Следует отметить, что Microsoft Word (или OpenOffice.org Writer, или LibreOffice Writer), по существу, тоже является WYSIWYG-редактором, хотя и «прячут от пользователя» свои команды разметки и форматирования внутри текстового документа.

К наиболее распространенным визуальным редакторам относятся, например, Microsoft FrontPage, Macromedia DreamWeaver, HotDog от Sausage Software и некоторые другие. Они имеют мощные средства визуального редактирования, одновременно предоставляя доступ к созданному HTML-коду. Таким образом, работа в этих редакторах может состоять из 2-х шагов:

1. с начала визуальном режиме создается нужный HTML-документ,
2. а затем, в случае необходимости, можно произвести «более тонкую» его настройку, редактируя полученный на первом шаге код.

Использование визуальных HTML-редакторов существенно экономит время и силы разработчиков Web-страниц.

Однако следует заметить, что ни один визуальный редактор не совершенен, и все они, так или иначе, ограничены в своих возможностях. Используя подобный редактор очень трудно создать что-либо оригинальное и эффектно смотрящееся, поскольку фантазия автора всегда будет ограничена рамками конкретной программы. А любому человеку всегда будет приятнее получить вещь, созданную мастером специально для него, чем приобрести аналог фабричного производства, который имеется дома у каждого второго. Поэтому для создания не «штампованных», оригинальных Web-страниц их код пишется руками при помощи текстовых HTML-редакторов, которые еще называются *редакторами тегов*.

Как правило, текстовые HTML-редакторы создаются небольшими фирмами или отдельными программистами, большинство из которых распространяется совершенно бесплатно. Среди таких редакторов есть совсем простые – например, WebCoder, созданный Дмитрием Захаровым. Редакторы тегов такого уровня, как правило, имеют различные функции, облегчающие написание кода. Так, например, при создании нового документа они позволяют быстро создавать заготовку будущей Web-страницы; по нажатию различных «горячих клавиш» и кнопок вставлять уже готовые конструкции (куски кода, спецсимволы и т. д.); автоматически вызывать браузер для просмотра редактируемого документа и т. п. Большинство из них так же поддерживает подсветку синтаксиса, т. е. HTML-код отображается различными цветами. Например, теги выделяются одним цветом, атрибуты – другим, их значения –

третьим, а непосредственно текст страницы – четвертым. Это значительно упрощает чтение и редактирование кода – открыв документ, легко можно найти по цвету нужный элемент и отредактировать его.

Есть и более совершенные текстовые HTML-редакторы, такие, например, как Macromedia HomeSite, HTMLPad FisherMan, HTMLPad 2007, QuickHTML, NotePad++ (только для Windows), Quanta Plus (для UNIX-подобных операционных систем), Bluefish, Geany (есть версии для UNIX-подобных операционных систем, и для Windows) и многие, многие другие. Они позволяют не только создавать довольно сложные элементы, такие, например, как таблицы, но и работать над проектами – наборами взаимосвязанных файлов, образующих сайт.

Между тем, поскольку HTML-документ является обычным текстовым файлом, то для его создания подойдет любой текстовый редактор. Простейший вариант – программа Блокнот, которая имеется у каждого пользователя, поскольку входит в стандартную поставку операционной системы Microsoft Windows. Она является наилучшим вариантом на начальном этапе освоения языка HTML – *с программы Блокнот когда-то начинали все создатели Web-сайтов*. Программа Блокнот всегда запустится по команде основного меню Windows **Пуск ⇒ Все программы ⇒ Стандартные ⇒ Блокнот**. Если при этом в нем набрать заготовку Web-страницы, представленную на Рис. 6.2, то она будет выглядеть следующим образом (Рис. 6.3):

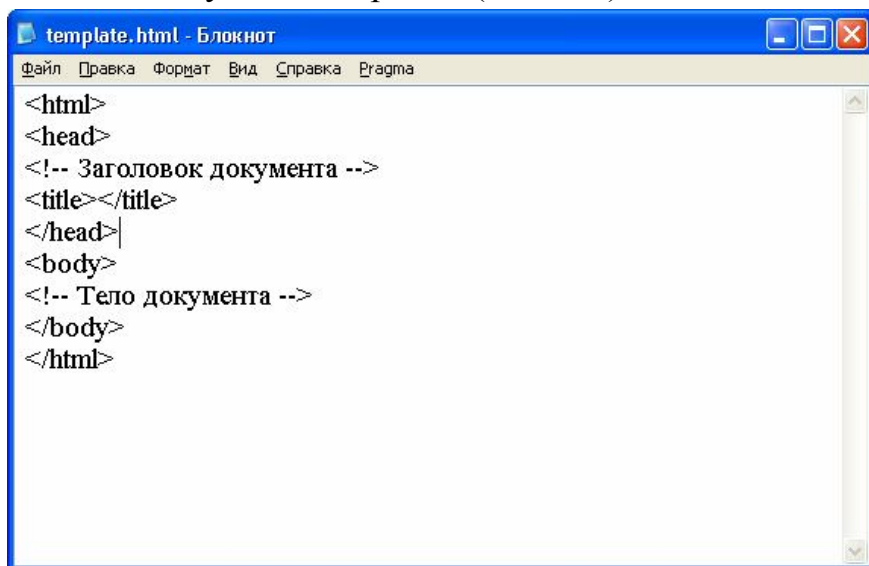


Рис. 6.3. Заготовка Web-страницы в программе Блокнот

⚠ При первом сохранении HTML- документа в окне диалога **Сохранить** необходимо явно указать:

1. тип файла – **Текстовый документ**,
2. имя файла – латинскими буквами и
3. расширение имени файла – **.html** или **.htm**.

В дальнейшем необходимо привыкать к латинским буквам в именах файлов, поскольку не все компоненты Интернет адекватно работают с русскими символами.

Если теперь двойным щелчком открыть созданный HTML-документ в браузере, например, Microsoft Internet Explorer, то будет выведена пустая страница (Рис. 6.4):

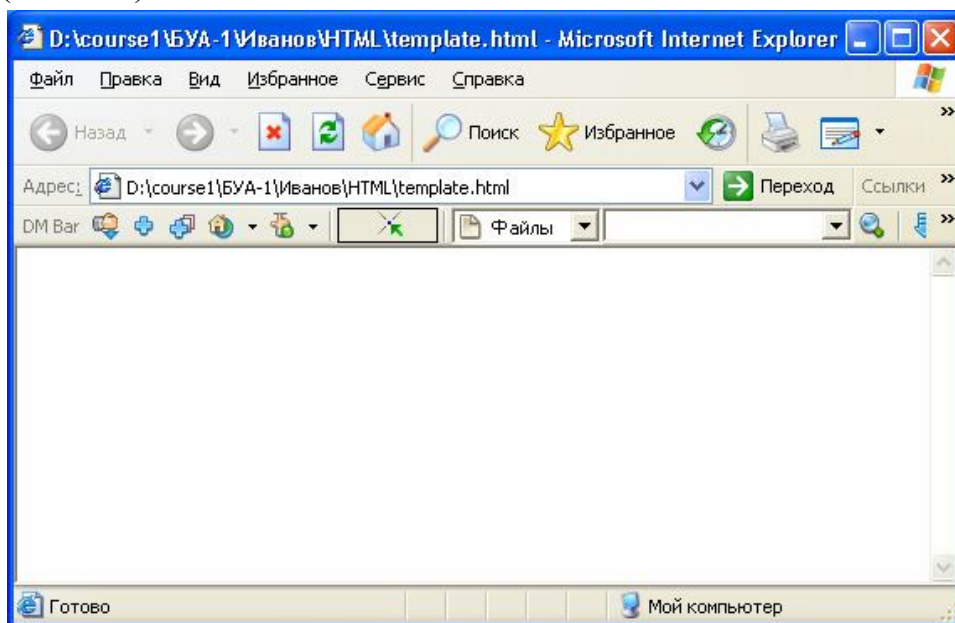


Рис. 6.4. Заготовка Web-страницы в браузере Microsoft Internet Explorer

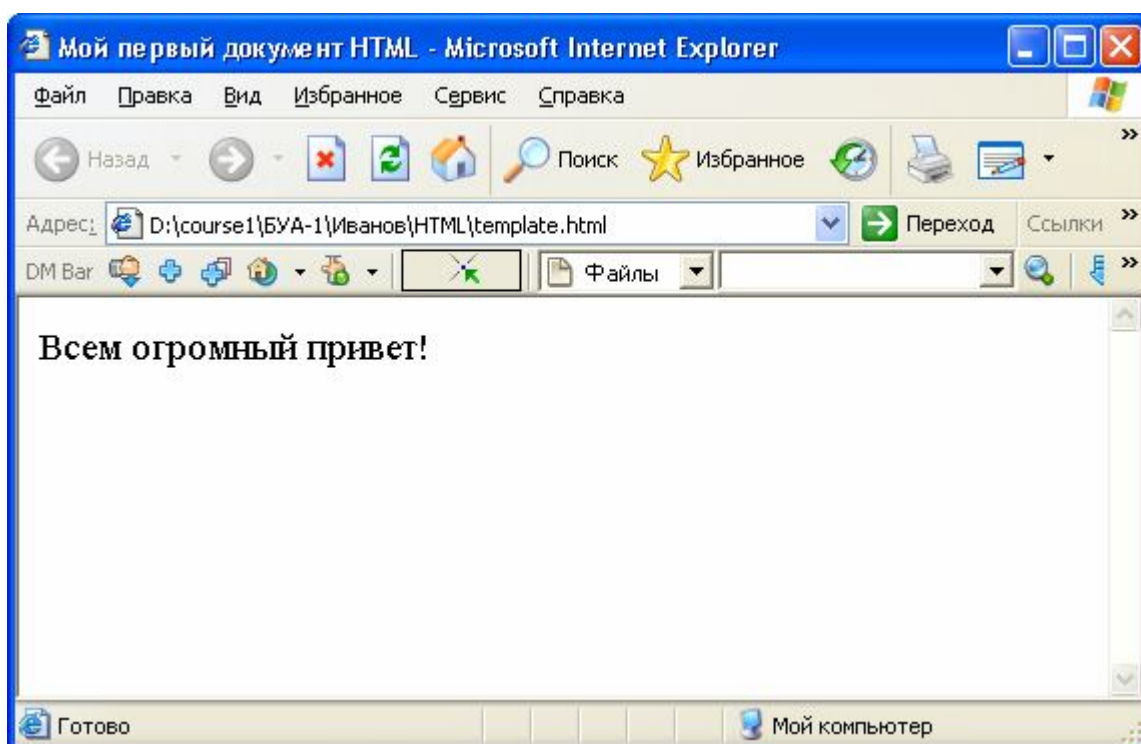
Для того, чтобы Web-страница приняла более информативный вид, необходимо в заготовку HTML-документа внести некоторые изменения. Например, между тегами <title> и </title> поместить текст «Мой первый документ HTML», а между <body> и </body> – «Всем огромный привет!». Тогда откорректированный текст HTML-документа должен выглядеть следующим образом:

```
<html>
<head>
<!-- Заголовок документа -->
<title>Мой первый документ HTML</title>
</head>
<body>
<!-- Тело документа -->
Всем огромный привет!
</body>
</html>
```

⚠ Для того, чтобы увидеть внесенные в HTML-документ изменения необходимо:

1. сохранить его в программе Блокнот (выполнив, например, команду **Файл ⇒ Сохранить**) и
2. обновить содержание окна браузера (щелкнув по кнопке **Обновить**).

В результате откорректированный HTML-документ должен выглядеть как на Рис. 6.5.



*Рис. 6.5. Первый HTML-документ*

Приведенные выше текст Web-страницы был разбит на строки только лишь с целью лучшей читабельности и восприятия человеком. Большинство современных браузеров воспринимают код HTML-файлов как одну длинную строку, пропуская при этом все лишние пробелы, символы конца строк и табуляции. То есть, «с точки зрения браузера» этот текст выглядит следующим образом:

```
<html><head><!-- Заголовок документа --><title>Мой  
первый документ HTML</title></head><body><!-- Тело  
документа -->Всем огромный привет!</body></html>
```

Если этот текст так же (без разбивки на строчки) набрать в Блокноте, то внешний вид Web-страницы в окне браузера при этом не изменится.

## СПИСОК ИСТОЧНИКОВ

1. Burks A. W., Goldstine H. H., von Neumann J. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. – Pt. I, vol. I, Institute for Advanced Study, Princeton, N. J., July 1946.
2. HTML в примерах А. Климова – <http://winchanger.narod.ru/webmaster.html>.
3. HTML справочник Владимира Городулина – <http://html.manual.ru/>.
4. Архитектура фон Неймана. Материал из Википедии – свободной энциклопедии – [http://ru.wikipedia.org/wiki/Архитектура\\_фон\\_Неймана](http://ru.wikipedia.org/wiki/Архитектура_фон_Неймана)
5. Берлинер Э.М., Глазырина И.Б., Глазырин Б.Э. Windows XP. Самоучитель. – М.: ЗАО «Издательство БИНОМ», 2002 г. – 416 с.; ил.
6. Бэрри Мане. Компьютерные сети: Пер. с англ. – М.: БИНОМ, 1995. – 400 с.
7. Вин Дж. Искусство web-дизайна. Самоучитель. СПб.: Питер, 2002. – 224 с.; ил.
8. Гук М.Ю. Аппаратные средства IBM PC. Энциклопедия. 3-е изд. – СПб.: Питер, 2006. – 1072 с.; ил.
9. Далхаймер К., Уэлш М. Запускаем Linux, 5-е издание. – Пер. с англ. – СПб.: Символ Плюс, 2008. – 992 с.; ил.
10. Дронов В.А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. – СПб.: БХВ-Петербург, 2011. – 416 с.; ил.
11. Евдокимов В.В. и др. Экономическая информатика. Учебник для вузов. Под ред. д.э.н. проф. В.В. Евдокимова. — СПб.: Питер, 1997.
12. Информатика (для технических специальностей): учебное пособие / Н.И. Иopa. – М.: КНОРУС, 2011. – 472 с.
13. Информатика для экономистов: Учебник / Под общ. ред. В.М. Матюшка. – М.: ИНФРА-М, 2007. – 880 с
14. Информатика для юристов и экономистов / Симонович С.В. и др. – СПб.: Питер, 2001. – 688 с.
15. Информатика и информационные технологии: учебное пособие / Ю.Д. Романова, И.Г. Лесничая, В.И. Шестаков, И.В. Миссинг, П.А. Музычкин; под ред. Ю.Д. Романовой. – 3-е изд., перераб. и доп. – М.: Эксмо, 2008. – 592 с. – (Высшее экономическое образование)
16. Информатика. Базовый курс / Под ред. С.В. Симоновича – СПб.: Издательство «Питер», 2000. – 640 с; ил.
17. Информатика. Теория и практика: Учеб. Пособие / В.А. Острейковский, И.В. Полякова. – М.: Издательство Оникс, 2008. – 606 с.; ил
18. Кирсанов Д. Веб-дизайн: книга Дмитрия Кирсанова. СПб: Символ-Плюс, 2006. – 376 с.; цв. ил.
19. Колисниченко Д.Н. Linux. От новичка к профессионалу. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2010. – 784 с.; ил.
20. Колисниченко Д.Н. Самоучитель Linux. Установка, настройка, использование. – 4-е изд., перераб. и доп. – СПб.: Наука и Техника, 2006. – 688 с; ил.
21. Колисниченко Д.Н. Самоучитель системного администратора Linux. – СПб.: БХВ-Петербург, 2011. – 544 с.; ил.
22. Коржинский С.Н. Настольная книга Web-мастера: эффективное применение HTML, CSS, JavaScript. Издание второе, исправленное и дополненное. – М.: Издательский торговый дом «КноРус», 2000. – 320 с.



23. Королев Л.Н. Архитектура электронных вычислительных машин. – М., Научный мир, 2005, 272 с., 65 ил.
24. Королев Л.Н. Структуры ЭВМ и их математическое обеспечение. – Наука, 1978.
25. Костромин В.А. Самоучитель Linux для пользователя. – СПб.: БХВ-Петербург, 2005. – 672 с.: ил.
26. Кролл Эд. Все об INTERNET: Руководство и каталог / Пер. с англ. С. М. Тимачева. – К.: Торгово-издательское бюро BHV, 1999. – 592 с.
27. Леонтьев В.П. Новейшая энциклопедия Интернета 2006. – М.: ОЛМА-ПРЕСС Образование, 2006. – 720 с.: ил. – (Новейшая энциклопедия)
28. Лесничая И.Г., Миссинг И.В., Романова Ю.Д., Шестаков В.И. Информатика и информационные технологии. Учебное пособие / Под ред. Романовой Ю.Д. – М.: Изд-во Эксмо, 2005. – 544 с. – (Высшее экономическое образование)
29. Лесничая И.Г., Миссинг И.В., Романова Ю.Д., Шестаков В.И. Информатика и информационные технологии. Учебное пособие / Под ред. Романовой Ю.Д. – М.: Изд-во Эксмо, 2005. – 544 с. – (Высшее экономическое образование)
30. Лоу Д. Компьютерные сети для «чайников». – К.: «Диалектика», 1995. – 256 с.
31. Майерс Г. Архитектура современных ЭВМ (в 2-х книгах). – Мир, 1985.
32. Макарова Н.В., Волков В.Б. Информатика: Учебник для вузов. – СПб.: Питер, 2011. – 576 с.: ил.
33. Мальчук Е.В. HTML и CSS. Самоучитель. – М.: Издательский дом «Вильямс», 2006. – 416 с.: ил.
34. Матросов А.А., Сергеев А.О., Чаунин М.П. HTML 4.0. – БХВ-Петербург, 2004. – 672 с.: ил.
35. Мюллер Дж. Оптимизация Windows XP. – СПб.: Питер, 2006. – 480 с.: ил.
36. Ногл М. TCP/IP. Иллюстрированный учебник – М.: ДМК Пресс, 2001. – 480 с.: ил.
37. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2007. – 958 с.: ил.
38. Поляк-Брагинский А. В. Локальные сети. Модернизация и поиск неисправностей. – СПб.: БХВ-Петербург, 2006. – 640 с.: ил.
39. Сетевые операционные системы / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер. 2002. – 544 с.: ил.
40. Степанов А.Н. Информатика: Учебник для вузов. 4-е изд. – СПб.: Питер, 2006. – 684 с.: ил.
41. Таненбаум Э. Архитектура компьютера. 5-ое изд. СПб.: Питер, 2013. – 848 с.: ил.
42. Учебник по Html (хтмл) для чайников Алленовой Натальи – <http://postroika.ru/html/>.
43. Фролов А.В., Фролов Г.В. Всемирная паутина. Ваш спутник в Интернете. – М.: Издательско-торговый дом «Русская Редакция», 2000. – 400 с.: ил.
44. Центр дистанционного обучения Харьковского национального университета городского хозяйства им. А. М. Бекетова – <http://cdo.kname.edu.ua/>.
45. Цифровой репозиторий Харьковского национального университета городского хозяйства им. А. М. Бекетова – <http://eprints.kname.edu.ua/>.
46. Экономическая информатика: / Под ред. П.В. Коноховского и Д.Н. Колесова. – СПб.: Питер, 2000. – 560 с.
47. Экономическая информатика: Учебник / Под ред. В.П. Косарева. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2004. – 592 с.: ил.

*Навчальне видання*

**ПОГРЕБНЯК** Борис Іванович  
**ВИСОЦЬКА** Галина Василівна

**КОНСПЕКТ ЛЕКЦІЙ**  
**ТЕХНІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
*(для студентів 5-го курсу денної та 6-го курсу заочної форм навчання  
магістрів спеціальності 8.18010013 – «Управління проектами»)*

*(рос. мовою)*

Відповідальний за випуск *М. І. Самойленко*

*За авторською редакцією*

Комп'ютерне верстання *К. А. Алексанян*

План 2013, поз. 115Л

---

Підп. до друку 12.06.2013р.

Формат 60x84/16

Друк на різнографі

Ум. друк. арк. 5,3

Тираж 50 пр.

Зам. №

---

Видавець і виготовлювач:

Харківський національний університет міського господарства імені О. М. Бекетова  
вул. Революції, 12, Харків, 61002

Електронна адреса: rectorat@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК №4064 від 12.05.2011